THE UNIVERSITY *of* LIVERPOOL

Agent-based Power System Protection and Information Management System of a Substation

Thesis submitted in accordance with the
requirements of the University of Liverpool
for the degree of Doctor of Philosophy

in

Electrical Engineering and Electronics

by

Chen Ma, B.Eng.

March 2009

Agent-based Power System Protection and Information Management System of a Substation

by

Chen Ma

To my parents

Acknowledgements

This thesis is dedicated to my parents to whom I owe everything. I must be indebted to them for their patience, understanding, encouragement, and self-giving love through the whole period of my life.

I have to thank my supervisor, Prof. Q. H. Wu, not only for his invaluable support and intellectual guidance, but also for his unwavering attention to precision, style and detail on research. Particularly, I really appreciate him for providing such a precious opportunity to enrich me on both an academic and personal level by his philosophy. His deep concern and anxiety about my daily life will never be forgotten.

I would like to thank Dr. W. H. Tang for patient discussion and support on my research project. I also want to thank my colleagues from Intelligence Engineering and Automation Research Group, the University of Liverpool, including Dr. J. Q. Feng, Dr. Z. Lu, Mr. Z. Yang, and Mr. D. Y. Shi for their unreserved assistance with many different problems I have encountered throughout my studies. Particularly, I must be grateful of Mr. J. Buse who gave me many valuable suggestions on this thesis.

My thanks also go to both the University of Liverpool for the studentship (2005 – 2008) and the Committee of Vice-Chancellors and Principals for the "Overseas Research Students Awards Scheme" (2005 – 2008).

Abstract

This thesis describes a multi-agent framework that is developed for the protection of distribution networks and substation information management. Within this framework, a number of software agents are derived based on a generic structure for performing various tasks. The generic structure defines an agent kernel which consists of a reasoning engine for decision making and three units for data and signal input, knowledge management and operation performing. An agent developed based on this structure can be easily modified and reconfigured in accordance with the change of its situated environments. Agent communications in this framework are based on the combination of FIPA-compliant protocols and the UDP protocol to support reliable and fast message transmission, respectively. Additionally, an agent-brokering mechanism is proposed within this framework to improve agent coordination in a large scale multi-agent system. A broker agent as a middleware agent is developed to manage the services offered by the provider agents and to communicate with the requester agents for taking requests and forwarding these requests to related provider agents.

Within the proposed multi-agent framework, an agent-based relaying scheme is investigated for the protection of distribution networks with distributed generation (DG) integrated. The existing protective relaying schemes are affected by the interconnection of DG, since it may change the amplitude, direction and duration of currents, reverse power flow and voltage profile, as well as reduce the fault impedance. The proposed agent-based relaying scheme provides a flexible and active solution to improve coordination between the protection relays and the integrated DG units. Specifically, a relay module and DG module

are derived consisting of the developed generic agents, which are able to connect with a protection relay and a DG unit, respectively, for receiving real-time operating signals and communicating with other modules. Using this scheme when the DG units are connected, the relays in their adjacent networks will be informed. Following the control principles maintained by the relay modules, the relay measurements and protection settings can be regulated.

In addition, an agent brokering-based anti-islanding scheme is proposed based on the multi-agent framework for preventing DG from islanded conditions. The relay modules and DG modules employed in the previous scheme are modified and the developed agent brokering mechanism is applied in this scheme to support coordination between these two modules. This scheme aims at transferring relay operating signals to its downstream DG modules by requesting the broker agent. According to the operational status of the upstream relays, the DG module is able to issues commands to the associated DG unit for controlling its connection status. Furthermore, in order to optimise the response time of the broker agent, a multi-brokering mechanism is proposed in this scheme that more than one broker agent is utilised for taking requests from multiple relay modules at the same time.

This thesis also focuses on managing a large amount of information in a substation using the multi-agent system (MAS) technology. Based on the proposed multi-agent framework, an agent-based substation information management system is developed. All of the generic agents derived within this framework are employed in this system for performing different tasks, including data acquisition, transformer condition assessment, device operation monitoring, data query, document retrieval, *etc.* In particular, the developed two agent-based protective relaying schemes can be also integrated within this system for information gathering. Additionally, this system is implemented in substation asset management. Three specific applications are investigated, including monitoring the protection status of relays and breakers, assessing transformer working conditions and information query.

In the thesis, descriptions of the simulations of the proposed agent-based relaying schemes, applications of the agent-based substation information management system, as well as the analysis of developed results are included to support the conclusion drawn in each work. Finally, a systematic summary is given, challenges are discussed and future research work is suggested.

Contents

List of Figures

List of Tables

List of Abbreviations and Symbols

Abbreviations

ACL	Agent Communication Language
AI	Artificial Intelligence
AID	Agent ID
AMS	Agent Management System
AOP	Agent-Oriented Programming
AP	Agent Platform
ARA	Attribute Reasoning Agent
ARM	Association Rule Mining
BDI	Belief Desire Intention
BIP	Brokering Interaction Protocol
CA	Communicative Act
CAL	Communicative Act Library
CB	Circuit Breaker
CCL	Constraint Choice Language
CL	Content Language
CNP	Contract Net Protocol
COM	Component Object Model
CORBA	Common Object Request Broker Architecture
CSCA	Cross Sensor Corroboration Agent
DAI	Distributed Artificial Intelligence
DF	Directory Facilitator
DG	Distributed Generation
DGA	Dissolved Gas Analysis

DSA	Distributed Situation Assessment
DVMT	Distributed Vehicle Monitoring Testbed
ER	Evidential Reasoning
FIPA	Foundation for Intelligent Physical Agents
FIPA SL	FIPA Semantic Language
GUI	Graphic User Interface
HF	High Frequency
HMI	Human Machine Interface
HTTP	Hypertext Transfer Protocol
IED	Intelligent Electronic Device
IIOP	Internet Inter-Orb Protocol
IP	Interaction Protocol
IRE	Identifying Reference Expression
JADE	Java Agent DEvelopment Framework
JAL	JACK Agent Language
JDBC	Java Database Connectivity
KIF	Knowledge Interchange Format
LAN	Local Area Network
LOG	Loss Of Grid
LOM	Loss Of Mains
MACE	Multi-Agent Computing Environment
MAS	Multi-Agent System
MICE	Michigan Intelligent Coordination Experiment
MKRA	Meta Knowledge Reasoning Agent
MTC	Message Transport Channel
MTP	Message Transport Protocol
MTS	Message Transport Service
NGT	National Grid Transco
OO	Object-Oriented
OMG	Object Management Group
PD	Partial Discharge
PRS	Procedural Reasoning System
PXI	eXtensions for Instrumentation
RDF	Resource Description Framework
ROCOF	Rate Of Change Of Frequency

TÆMS	task analysis, environment modeling, and simulation
TFP	Apriori-Total From Partial
TOC	Time Overcurrent
W3C	World Wide Web Consortium
XML	Extensible Markup Language
VVS	Voltage Vector Shift

Chapter 1

Introduction

This thesis is concerned with the development of distribution network protection and substation information management for power system automation using the multi-agent system (MAS) technologies. An overview of power system automation concerning protection and information management and a brief introduction to the MAS technologies are given at the beginning. The existing problems and deficiencies of using the conventional methodologies to handle the issues of protection and information management in a power system are discussed in detail. Moreover, the motivations of this thesis are presented followed by a description of the contributions of this research work. Finally, a thesis outline is provided to give a clear view of the entire contents.

1.1 Background of Power System Automation

The purpose of a power system is to generate, transmit and distribute electrical energy to consumers. Basically, power system automation is an act that controls a power system via automated processes within computers, intelligent instrumentations and control devices [1]. These processes, relying on data acquisition, power system supervision and protection, condition monitoring, information management, *etc.*, work together in a coordinated automatic fashion with both reliability and economy. In this section, two important issues, protection and information management, are introduced.

1.1.1 Power system protection

Protection principles

One of the most important part of electrical power engineering is to protect an electrical power system from faults by isolating faulted parts quickly and to minimise the shock to the rest of the system as much as possible [2]. Generally, a power system is divided into protection zones defined by the equipment and the available circuit breakers. Six categories of protection zones are possible in each power system, *i.e.,* generators and generator-transformer units, transformers, buses, lines, utilisation equipment (motors, loads and others) and capacitor or reactor banks. In Figure 1.1, most of the zones are illustrated. Within this context, there are four major principles for power system protection.

- Reliability: reliability has two aspects, dependability and security. The dependability of a protection system is "the ability to detect and disconnect all faults within the protected zone", while security is the "ability to

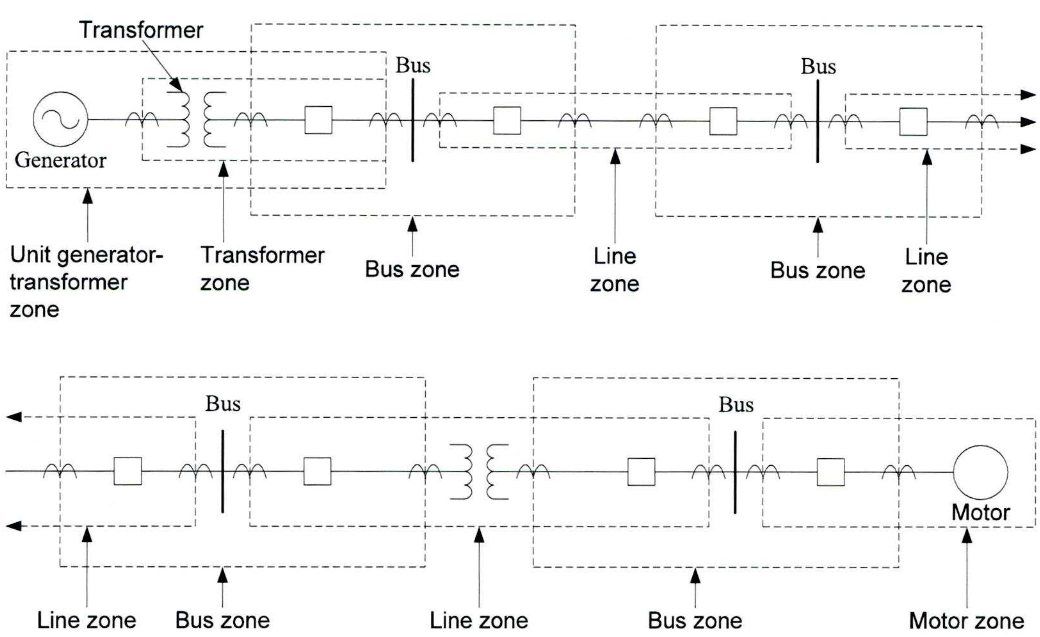


Figure 1.1: Typical relay primary protection zones in a power system.

reject all power system events and transients that are not faults so that healthy parts of the power system are not unnecessarily disconnected" [3].

- Selectivity: a protection system should only disconnect the faulted parts or reduce the parts containing faults in a power system to minimise fault consequences [4].

- Speed: it is desirable that a protection system isolates a fault zone as rapidly as possible, but zero-time or very high speed protection may result in an increased number of undesired operations. Therefore, time remains as one of the best means of distinguishing between tolerable and intolerable transients.

- Economics: the fundamental of economics is to obtain the maximum protection for the minimum cost and to balance the reliability and expense of a protection system

Protection relay

A power protection system is completely arranged by a number of protection devices (*e.g.*, protection relays, circuit breakers, reclosers, fuses, *etc.*,) required to achieve specified functions based on the protection principles to minimise the impacts of the unavoidable faults in the system. In order to fulfill these requirements with the optimum speed for many different configurations, operating conditions and construction features of a power system, it has been necessary to develop many types of protection relays that respond to the various functions of the power system quantities [5].

Relays can be classified in many different ways, such as by function, input, performance characteristics, or operating principles [2]. Classification by function is most common. There are five basic functional types, *i.e.,* protective, regulating, reclosing and synchronising, monitoring, and auxiliary. Specifically, protective relays can be applied to all parts of a power system, *e.g.,* generators, buses, transformers, transmission lines, distribution lines and feeders, motors, *etc.* Furthermore, they classified by input are known as current, voltage, power, frequency and temperature relays.

It is difficult to evaluate the performance of an individual relay, because many relays near the trouble area may begin to operate for any given fault. The primary relays operate to isolate the fault area and all other altered relays are return to their normal quiescent mode. In particular, relay operation can be categorised as three types, *i.e.*, correct operation, incorrect operation and no conclusion. Correct operation indicates that at least one of the primary relays operated correctly, none of the backup relays operated to trip for the fault and the fault area was properly isolated in the time expected. Incorrect operations result from a failure, a malfunction or an unanticipated or unplanned operation of the protection system, which are caused by misapplication of relays, incorrect relay settings, personnel errors and equipment problems and failures. Moreover, no conclusion refers to circumstances during which one or more relays have or appear to have operated, such as the circuit breaker tripping, but no cause can be found.

1.1.2 Substation information management

In view of the pressure on cost reduction and productivity improvement, which is caused by deregulation and unbundling vertical structured utilities into smaller dedicated entities for generation, transmission and distribution, the greatest changes may happen in a substation [6]. Specifically, three key areas are expected to be developed concerning these changes, which are listed as follows:

- access to substation information;
- operation and maintenance strategies; and
- asset management.

Open access to substation information is a way of quickly providing benefits to the business. The application of intranet technology, will allow information relating to the condition and performance of the primary plant, such as circuit breakers and transformers, to be released from the closed substation information systems where it is currently held. The processing of this information will improve asset management by optimising the maintenance of the high value primary plant and in the future assisting with replacement decisions.

1.1.3 Power system asset management

Another field of power system automation focused by this thesis is asset management. Briefly speaking, asset management is the combination of management, finance, economy, engineering and other practices applied to physical assets, with the objective of providing the required level of service in the most cost effective manner [7]. Basically, an asset is regarded as an item or a property owned by an individual or business who has monetary value, which can be identified into three types, including physical assets (*e.g.* equipments), financial assets (*e.g.* financial instruments and equity accounted investments) and intangible assets (*e.g.* operating licence, knowledge and skills of staff).

Additionally, power system has placed unprecedented strains and dilemmas with the conflicting objectives on asset utilisation due to the pressure from both massive industrial growth and capital expenditure increasing. Many electric power industrials have been challenged by the potential risks of increased unreliability of the aging key assets. Therefore, asset management is now in the minds of many system operators in the electric power industry and many efforts have been made to implement asset management into power system [8] [9] [10]. The main purpose of the asset management in power system is to manage the physical assets, such as transformers, generators, *etc.*, and their associated performance optimally by integrating technical diagnosis and management decisions. Furthermore, the main elements of a typical asset management system for power system are summarised as follows:

- Proactive maintenance with optimised repairing strategies.
- Condition monitoring and fault diagnosis.

- Risk evaluation of system operations.
- Information management and knowledge representation.
- Life-cycle cost analysis and cash flow prediction.

1.2 Introduction to Agent and Multi-agent System

1.2.1 Definitions

The concept of an *agent* can be traced back to the early days of research into Distributed Artificial Intelligence (DAI) in the 1970s. Carl Hewitt developed an actor model [11] that is a self-contained, interactive and concurrently-executing object, possessing internal state and communication capability. The object, firstly regarded as a computational agent, had a certain encapsulated internal state and could respond to messages from other similar objects.

Broadly speaking, the history of agent development can be splitted into two main strands: the development of smart agents and software agents. The first strand working on smart agents from 1977 concentrated on macro issues, for instance, the interaction and communication between agents, the decomposition and distribution of tasks, coordination, cooperation and negotiation of systems, *etc.* The major objectives of these researches are to specify, analyse, design and integrate systems comprising of multiple collaborative agents [12]. For example, MACE (Multi-Agent Computing Environment) [13], a programming environment, supports experimentation with different styles of DAI systems, at different levels of complexity. The MICE (Michigan Intelligent Coordination Experiment) testbed [14] as an extension of previous experimental systems allows an experimenter to specify the constraints and characteristics of an environment in which agents are simulated to act and interact. TÆMS (Task Analysis, Environment Modelling, and Simulation) [15] as a framework models the complex computational task environments that allow a user to both analyse and quantitatively simulate the behaviour of a single or multi-agent systems with respect to interesting characteristics. DRESUN [16] for research on distributed situation assessment (DSA) is developed to explore the implications of having agents with more sophisticated evidential representations and control capabilities than the agents that were used in our earlier research with the Distributed Vehicle Monitoring Testbed (DVMT). ARCHON [17], a framework for intelligent cooperation, can be used to facilitate cooperative problem solving in industrial applications.

In addition, since 1990 there has evidently been another distinct strand to the research and development worked on software agents. Wooldridge and Jennings firstly proposed an intelligent agent concept [18] [19] which complemented and broadened the typology of agents being investigated by agent researchers. Russell and Norvig extended an intelligent agent to a rational agent [20] that is to take actions based on information from and knowledge about the agent's environment. The rational agent tends to maximise the chances of success using commonly accepted logical inference rules.

Intelligent agent

Generally, there is no fixed definition of an agent or an intelligent agent. The *agent* concept is typically used to refer to software components that have their own thread of control and hence may act autonomously, and are capable of sensing and reacting to changes in some environments. Moreover, intelligent agents have other properties, such as the ability to communicate with other agents. Recently, intelligent agents have become widely used in the modelling of complex and distributed problems [21]. One of the most widely used definitions of an intelligent agent is proposed by Wooldridge and Jennings in 1995, which defines an agent as the following properties [18]:

- Autonomy: intelligent agents are able to operate without intervention of humans and have some kind of control over their actions and internal state;

- Social ability: an intelligent agent can negotiate and interact with other agents via an agent communication language, which allows agents to converse rather than simply pass data.

- Reactivity: an agent is capable of perceiving the changes in its environment, taking actions based on those changes, and responding in a timely fashion.

- Pro-activeness: an intelligent agent dose not simply act in response to its environment, it is able to exhibit goal-directed behaviour by taking the initiative, which means an agent will dynamically change its behaviour in order to achieve its goal.

Multi-agent system

Ferber defines a multi-agent system as a system consisting of an environment, a set of objects which exist in that environment and can be acted upon by agents and a set of agents, which represent the "active entities" of the system [22]. Basically, a multi-agent system is simply regarded as a system comprising two or more software agents with no overall system goal. Each separate agent is related to objects by relations and acts on those objects by means of operations.

Additionally, Ferber defines two extreme classes of agent in a multi-agent system [22], a purely communicating agent, which has no physical environment and acts only by communicating with other agents, and a purely situated agent, which has no communication with other agents, but is situated in a physical environment and acts through that environment. Many multi-agent systems also employ agents that have features of both types. Depending on Wooldridge's definition of an intelligent agent, agents in a multi-agent system are capable of communicating directly with each other.

1.2.2 Agent architectures

Reactive architectures

An example of a reactive architecture for building agents is proposed by Brooks in [23], called subsumption architecture which does not employ an explicit knowledge representation. the subsumption architecture is a hierarchy of task-accomplishing behaviours. In recent papers [24] [25], Brooks pointed three key issues of the subsumption architecture:

1. Intelligent behaviour can be generated *without* explicit representations of the kind that symbolic AI proposes.
2. Intelligent behaviour can be generated *without* explicit abstract reasoning of the kind that symbolic AI proposes.
3. Intelligence is an *emergent* property of certain complex systems.

A number of robots were built based on the subsumption architecture with lower layers representing simpler behaviours, which have a high priority, and higher layers representing more abstract behaviours, and having lower priority. Another sophisticated approach was developed by Rosenschein and Kaelbling [26]. In their situated automata paradigm, an agent is specified in terms of a logic of knowledge, which is compiled down to a low-level digital machine to satisfy the intentional specification.

Hybrid architectures

Many researchers have suggested that the reactive approach is not suitable for building agents [27]. They have argued the case for hybrid systems, which attempt to marry classical and alternative approaches.

A well-known cognitive architecture is the Belief-desire-intention (BDI) [28] [29], in which the agent's knowledge base is described by a set of *beliefs,* (those facts which an agent considers to be true) *desires* (those conditions which the agent wishes to bring about) and *intentions* (actions which the agent has committed to perform). These are explicitly represented in the knowledge base; for example, the Procedural Reasoning System (PRS) implementation [29] represents beliefs and goals as ground literals (sentences containing no implications, binary operators or variables) in first-order logic [30]. As described in [30], a BDI agent is capable of both reactive and deliberative behaviour. On each execution cycle of the interpreter, the agent retrieves new events from the environment. It then generates a set of *options*, which are plans or procedures that the agent is capable of carrying out, both in response to events and in order to achieve its goals. The agent will then execute, or partially execute, one or more of the selected options. This process is repeated for the agent's lifetime.

Learning-based architectures

Learning-based architectures, such as reinforcement learning [31], genetic programming [32], or inductive logic programming [33], are used to enhance agent performance. While it is possible to use learning to improve the capabilities of an agent using an architecture, such as BDI, it is also common to incorporate learning into a much simpler agent architecture. For example, [34] used machine learning methodologies to recognise plans being undertaken by other agents in a BDI architecture uses case-based reasoning in a BDI agent for information retrieval. Learning agents have been applied in a number of domains, including user interfaces [35], telecommunications [36], control and robotics [37].

Layered architectures

Layered architectures such as TouringMachines [38] and INTERRAP [39] are cognitive architectures consisting of one or more layers. Suggested by Ferguson [38], the advantage of a layered architecture is that a layered agent, having different levels of behaviour operating concurrently, is capable of reacting to the changes of circumstances while planning its future actions and reasoning about the behaviour of other agents.

In TouringMachines, all three layers are connected to the agent's sensors and effectors. The three layers operate concurrently and are unaware of each other, while a control mechanism is used to filter the inputs and outputs and prevent conflicts. In INTERRAP, the sensors and effectors are connected only to the lowest layer (the behaviour-based layer). Activation requests are passed upward through the layers, and commitments are passed downwards. Unlike the subsumption architecture (which is a form of layered architecture), both TouringMachines and INTERRAP are based on explicit knowledge representation [38].

1.2.3 Standards for agent development

Another question is how to design an agent. Currently, several organisations having developed or aiming to develop standards related to the interoperation and interaction of intelligent software entities, such as FIPA, OMG, the Global Grid Forum, and the World Wide Web Consortium. This section introduces the standards for agent development. In Figure 1.2, the timescales of recent developed agent standards are summarised.

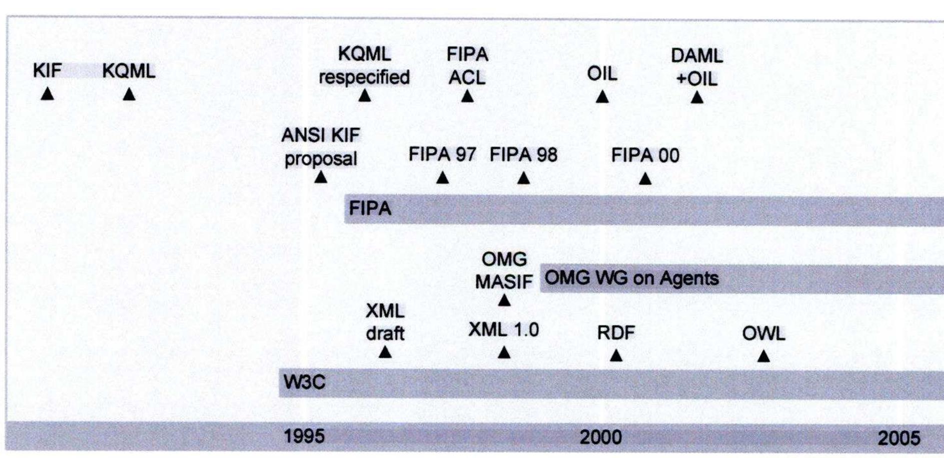


Figure 1.2: Timescales for the development of agent standards.

As one of the leading international organisations for the development of agents and multi-agent systems, FIPA is dedicated to promoting the industry of intelligent agents by openly developing specifications that support interoperability among agents and agent based applications. FIPA standards were originally proposed in 1996 to form the specifications of software standards for heterogeneous and interacting agents and agent based systems [40]. In the past a few years, FIPA has been widely recognised as the major standards in the area of agent-based computing. A number of standard specifications have been developed by FIPA, such as Agent Communication Language (ACL), Interaction Protocols (IPs), *etc.* Figure 1.3 gives an overview of the FIPA agent development standards. Four parts are included in FIPA standards, an abstract architecture, agent management, agent communication and agent message transport, which will be introduced in detail in this section.

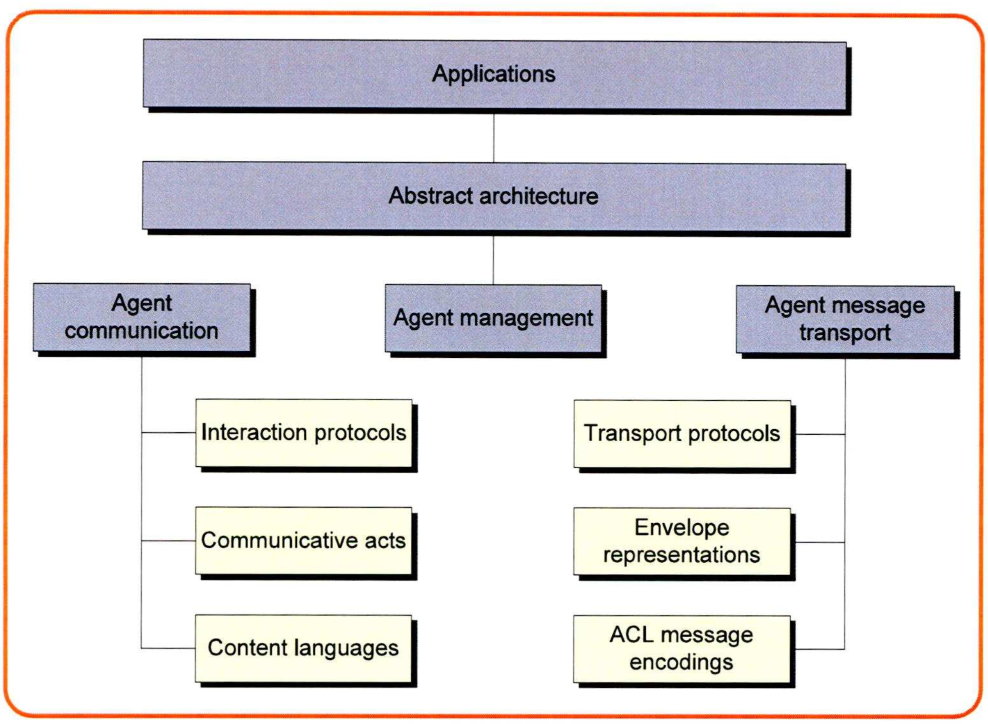


Figure 1.3: Overview of the FIPA standards.

FIPA abstract architecture

The FIPA abstract architecture specification (SC00001L)[1] acts as an overall description of the FIPA standards for developing multi-agent systems. The aim of the FIPA abstract architecture is to develop semantic meaning message exchange between the different agents. It includes the management of multiple message transport and encoding schemes and locating agents and servers via directory services. Figure 1.4 demonstrates the FIPA abstract architecture mapped to different concrete realisations.

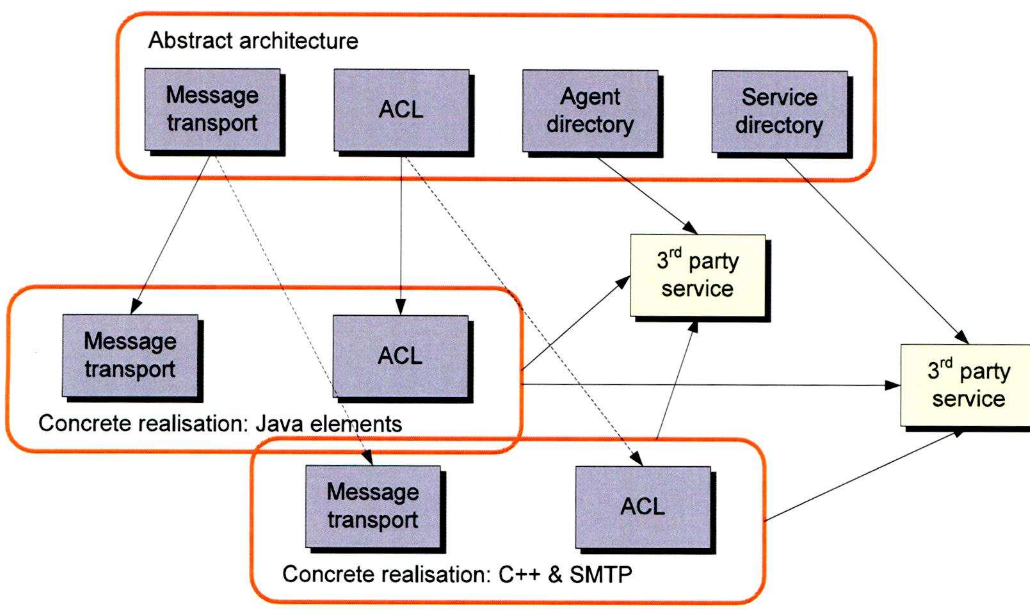


Figure 1.4: FIPA abstract architecture mapped to different concrete realisations.

In addition, it also supports mechanisms to create the multiple concrete realisations for inter-operation. The scope of this architecture includes the following items[41]:

- a model of services and discovery of services available to agents and other services;

[1]All the specifications mentioned in this section are obtained from the website of FIPA organisation. `http://www.fipa.org/repository/standardspecs.html`

- the message transport inter-operability;
- supporting various forms of ACL representations and content languages;
- supporting the representations of multiple directory services.

FIPA agent management system standards

The FIPA agent management system specification (SC00023K) denotes an agent management reference model of the runtime environment that FIPA agents inhabit. The logical reference model is established for agent creation, registration, communication, location, migration and retirement [41]. The reference model includes a set of logical-based entities, such as:

- an agent runtime environment for defining the notion of agenthood used in FIPA and an agent life cycle;
- an Agent Platform (AP) for deploying agents in a physical infrastructure;
- a Directory Facilitator (DF) which provides a yellow pages service for the agents registered on the platform;
- an Agent Management System (AMS) acting as a white pages service for supervisory control over access to the agent platform; and
- a Message Transport Service (MTS) for communication between the agents registered on different platforms.

The construction of the FIPA agent management reference model is illustrated in Figure 1.5.

FIPA agent message transport service

The FIPA agent message transport service specification (MTS) (SC00067F), as part of the FIPA agent management specification, supports the message transportation between the inter-operating agents. Two major specifications

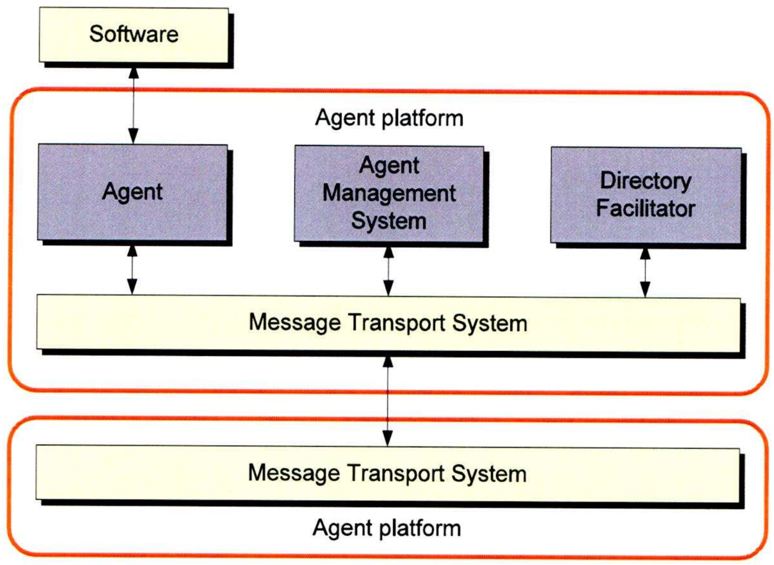


Figure 1.5: FIPA agent management reference model constitution.

are involved, *i.e.*, a reference model for an agent MTS and the definitions for the expression of message transport information to an agent MTS [41].

A three layered reference model is provided by MTS, *i.e.*, the message transport protocol (MTP) for physical messages transferred between two agent communication channels (ACCs), the MTS which provides the FIPA ACL messages transportation between agents on the platform, and the ACL representations from both MTS and MTP. Figure 1.6 shows the FIPA message transport reference model.

Additionally, other distinct components are involved into an agent MTS, which are:

- two transport protocols, for transporting messages between agents using the Internet Inter-Orb Protocol (IIOP) and Hypertext Transfer Protocol (HTTP), specified by FIPA Message Transport Protocol for IIOP (SC00075G) and FIPA Message Transport Protocol for HTTP (SC00084F) respectively;
- two message transport envelope specifications, *i.e.*, FIPA Agent Message Transport Envelope Representation in Extensible Markup Language

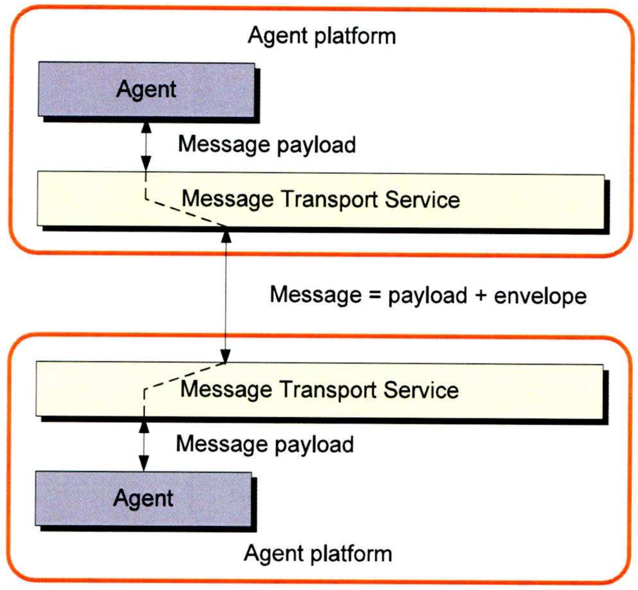


Figure 1.6: FIPA message transport reference model.

(XML) Specification (SC00085J) and FIPA Agent Message Transport Envelope Representation in Bit-Efficient Encoding Specification (SC00088D), which provide syntactic representations of a message envelope in XML form and bit-efficient form, respectively;

- three message representation specifications, *i.e.,* FIPA ACL Message Representation in Bit-Efficient Encoding Specification (SC00069G), String Specification (SC00070I) and XML Specification (SC00071E) for representing ACL syntax in a bit-efficient form, string form and XML form, respectively.

FIPA agent communication standards

One of the most important areas that FIPA standardised is agent communication which is the core category at the heart of the FIPA multi-agent system model. Four components are involved in FIPA Agent Communication specifications, *i.e.,* Agent Communication Language (ACL) Message, Interaction Protocols (IPs) of message exchange, speech act-based Communicative Acts

(CAs) and Content Language (CL) representations.

- FIPA ACL Message Structure specification (SC00061G) standardises the form of a FIPA-compliant ACL message structure to ensure inter-operability;
- A number of different interaction message exchange protocols are dealt by FIPA IPs specifications, such as request and query interaction protocols, brokering and recruiting interaction protocols, subscribe and propose interaction protocols, *etc*;
- FIPA Communicative Act Library (CAL) specification (SC00037J) defines the structure of the CAL and the formal basis of FIPA ACL semantics;
- A set of languages used in FIPA Messages are denoted by FIPA CLs as listed as follows:
  - a concrete syntax for the FIPA Semantic Language (FIPA SL) is defined by FIPA SL Content Language specification (SC00008I) for use in conjunction with the FIPA ACL;
  - FIPA Constraint Choice Language (CCL) Content Language specification (XC00009B) allows agent communication to involve exchanges about multiple interrelated choice;
  - FIPA Knowledge Interchange Format (KIF) Content Language specification (XC00010C) expresses the objects and propositions as terms and sentences, respectively.
  - FIPA Resource Description Framework (RDF) Content Language specification (XC00011B) constructs components of FIPA SL in the resource description framework representation.

1.3 Applications of Multi-agent Systems in Power System Automation

For over a decade, the MAS technology has been developed for a range of power system applications, including condition monitoring [42], diagnostics, power system restoration [43], market simulation [44] [45], network control [46], and automation [47]. Moreover, the technology is maturing to the point where the first multi-agent systems are now being migrated from the laboratory to the utility, allowing industry to gain experience in the use of MAS and also to evaluate their effectiveness [43].

1.3.1 Advantages

MAS have a tendency to be exploited in two ways: as an approach to build the flexible and extensible hardware/software systems; and as a modelling approach for representing a real-world situation of interacting entities, and giving a way of testing how complex behaviours may emerge. The potential benefits of applying multi-agent system in power engineering summarised in [48] are listed as follows:

- Flexibility and extensibility: MAS has the ability to respond correctly to dynamic situations, and support for replication in varied situations, which is beneficial for many power engineering application areas. MAS should be able to select the most appropriate action from a number of possible ones, such as the ability to construct a new plan if a particular control action fails. Furthermore, extensibility connotes the ability to easily add new functionality to a system, augmenting or upgrading any existing functionality. For instance, a distributed network control and management system responsible for voltage control may be extended to also automate restoration and the management of distributed generation.
- Open architecture: MAS in the early ages tended to be closed architectures, with one set of agents whose communication explicitly defined

by the system creator being deployed every time the system was run. However, this format limits the benefits of using newly created agents, since the existing agents would have no way of locating them and communicating with them. Therefore, newly developed MAS often takes an open architecture into account, setting no restrictions on the programming language or origin of agents joining the system. This is achievable through adherence to messaging standards, one example is a set of standards for an open architecture defined by the FIPA [49]. Within this agent platform, there is no restriction on the creation and messaging of agents, moreover, locating particular agents or agents offering particular services within the platform is achievable.

- Platform for distributed systems: An agent is distinct from its environment, although it may impacts upon which actions an agent performs. In practice, the agent platform supports the distribution of agents across a network: each computer runs a platform and agents are deployed within the platform. There is no difference between agents running on the same computer or on different computers, as the instances of the platform running on separate machines could seamlessly connect to each other.

- Fault tolerance: One of the standard engineering approaches to gaining fault tolerance is to build redundancy into systems, such as providing more than one agent with a given set of abilities. In practice, there are three ways to construct a MAS with fault tolerance. One of the approaches is a simple duplication of each agent, possibly with distribution of duplicates across different computers. This has been proved effective in a network with physical faults. However, tolerance to programming-related faults would require a design-intensive solution, such as coding two different agents with the same functionality. Thirdly, requirements of different applications differ in terms of levels of fault tolerance, the approach could also be application-specific. Moreover, the functionality can be enhanced by the flexibility offered by an open architecture.

- Modelling approaches: As an open architecture system, MAS provides an ideal platform for modelling approach. Inspired by the previously invented object-oriented (OO) systems design, natural representation of the world has also been considered in the implementation of MAS. The main benefit of OO is data-encapsulation, in which the particular data structures used to hold attributes of an object are hidden from external objects, but are indirectly accessible through methods calls and standard interfaces. MAS inherits this feature from OO, however, on top of it, it offers a higher level of abstraction by making the "methods" (actions) of an agent hidden. They have the similar accessibility with the internal data structure that both of them can only be indirectly accessible through standard messaging interfaces.

1.3.2 Distributed control

A number of researchers are considering agent-based approaches as an alternative to centralised power system management and control [46] [50]. Distributed control requires agents that are capable of a range of actions, such as monitoring local conditions, controlling switchgear and other plant, and coordinating with other regions of the network.

For example, the ARCHON system [51] described earlier was used to perform fault identification and service restoration in a power transmission network, and was installed in an actual control centre. Seven agents were used, based on both existing and new expert systems in the control centre. Each agent was responsible for a particular task, for example, blackout area identification or control system interface. The application of ARCHON described was used only in the control centre, and was not a full substation automation system. However, the general principles of the ARCHON system, including the use of wrappers, may be applied to the design of such a system.

1.3.3 Power system restoration

Power system restoration and protection is an area where the analogue between agents and protective devices is being explored [52] [53] [54] [55]. In all the papers above, protection relays and associated equipment are regarded as agents and their functionality augmented accordingly. In this case, MAS technology was under investigation as a way of developing novel restoration and protection schemes for fault tolerant and self-coordinating.

Baxevanos and Labridis [56] extended research upon the potentials of implementing distributed artificial intelligence technology to achieve high degrees of independency in distribution network protection and restoration processes. A multi-agent system is applied to control switching operations in a power system and to justify the need to distribute activities in contradiction to the centralised methodologies. A proper model of the real environment is introduced in order to define the designing parameters of a prototype agent entity, which is a part of a cooperative network-management system. The system's goal is to autonomously perform effective fault management upon medium-voltage power distribution lines. The structure of the agent entity was described by means of the agent behaviours being implemented and the system was implemented and tested in simulation.

1.3.4 Condition monitoring

Another key application area for MAS in power engineering is the management and interpretation of data for a wide variety of power engineering monitoring and diagnostic functions [48]. A typical example is the transformers, which have a range of sensors used to monitor them. MAS allows the combination of data from all these sources in a flexible manner by delegating each of the different tasks to an autonomous agent. In this case, each agent can be informed by each other if any significant fault has been detected. As mentioned previously, the property of flexibility allows the integration of as much information as is currently available.

Mangina *et al* [57] have developed a condition monitoring multi-agent system (COMMAS) using three layers of agent, which has been used for condition monitoring of power plants [58]. In this system, Attribute Reasoning Agents (ARAs) monitor and interpret sensor data, Cross Sensor Corroboration Agents (CSCAs) combine data from different sensors, and Meta Knowledge Reasoning Agents (MKRAs) provide diagnostics based on the information provided by the other agents. By reasoning about what causes a gas turbine to move from one state to another, the agents are able to identify the causes of faults.

1.3.5 Information management

MAS technology is an excellent tool for collecting and manipulating distributed information and knowledge [48]. For example, Lucas *et al.* [59] described a multi-agent system for document management, querying and decision making for the Iranian power industry. The system consists of interface agents which is able to interact with the users, resource manager agents for representing information resources, and coordinator agents which take requests from interface agents and pass them to the appropriate resource manager agents. These agents are used to connect several databases and sets of documents, including the Internet.

1.3.6 Other applications

In addition to the applications introduced above, MAS can be used in other areas of power engineering. For instance, the system described in [60] uses negotiation between two agents, representing an independent system operator and a transmission company, to determine whether a transmission circuit may be operated in excess of its rated load. The agents exchange proposals until a mutually acceptable solution is reached. The system was implemented in Java and tested on a simulator. It was also used for decision making in other aspects of a power system [61].

Additionally, the use of agent systems as a modelling approach is beneficial to the simulation of complex power systems, energy markets, overall energy networks, and energy utilisation, since within modern power systems, several operations are too complicated to model and simulate using traditional methods [48]. MAS was applied as an approach for simulating the energy marketplace, where agents model suppliers, brokers, generators, and customers [62] [45]. More recently, MAS technology has been suggested for the integration and coordination of different models and modelling software packages [63].

1.4 Problem Statement

The traditional power automation systems have a number of drawbacks. For example, the interoperability of the power devices is hampered by an excess of incompatible hardware interfaces and protocols [64], and more access to power system information is required in order to make the correct decisions. Concerning the protection, information management and asset management introduced in the previous section, the problems existed in the traditional power automation systems are addressed as follows:

- The interconnection of Distributed Generation (DG) with the existing power network raises up many problems, especially for the protection issues. DG affects the amplitude, direction and duration of the fault currents, reverses power flow and voltage profile and reduces the reach area of an impedance relay. Misoperations of a protection relay may appear, because the traditional relaying schemes are based on a pre-defined procedure that has to be set up case-by-case with the fixed parameter settings determined in manufactory. Moreover, the existed relaying schemes assume that the topologies of the power network are unchanged and the transmission power varies only within a small boundary. Therefore, the traditional relaying schemes may not satisfy the requirements of protecting a DG integrated power network.

- Islanding operations of DG always cause many serious impacts on protection, operation, and management of distribution systems. The existing islanding detection methods which measure voltage magnitude, phase displacement and frequency change can be only utilised if there are large changes in loading for DG after the loss of the main power supply. However, in case of small changes appear, these methods can not make proper detections. Furthermore, even the existing communication-based islanding detection methods, such as transfer tripping scheme, support for high-speed and exact islanded condition detection, it still can not be applied in large scales, since they require specific communication cables and special environments, such as optic fiber or microwave, which are very expensive and complicated.

- The increasing use of the intelligent electronic devices (IEDs) and the networks in power automation systems has led to the availability of a large amount of data and various types of information. It is difficult to effectively manage and to convert the data into knowledge to enable engineers to make use of it. Furthermore, the current power automation systems are inflexible and cannot easily accommodate new requirements or changes to the substation plant and the monitoring equipments.

Therefore, a framework is required to provide open access to power system information via the power company's local area network (LAN) and wide area network (WAN) and to integrate previously separate functions such as protection, condition monitoring, fault diagnosis and information management. In this thesis, a framework to achieve this objective is proposed using the MAS technology.

1.5 Motivations

As mentioned above, this thesis is devoted to develop protection, fault diagnosis and asset management for power system automation using MAS technology. In comparison with the client-server and the object-oriented systems, multi-agent systems have several claimed advantages. Jennings [21] stated that "the natural way to a complex system is in terms of multiple autonomous components that can act and interact in flexible ways in order to achieve their set objectives", and also that agents provide a "suitable abstraction" for modelling systems consisting of many subsystems, components and their relationships. Ferber described in [22] that how agents, as a form of distributed artificial intelligence, are suitable for use in application domains which are themselves widely distributed. The modern power system, with substations distributed throughout a wide area, falls into this category of systems.

Therefore, a multi-agent framework has been proposed for developing the protection of distribution networks and substation information management. A number of software agents are developed based on the generic structure, which are easy to be modified and reconfigured in a multi-agent system according to it situated environments. An agent platform is established for managing agent executions and communications. Moreover, within this framework, an agent brokering mechanism is developed to improve coordination among the agents.

Based on the this framework, an agent-based relaying scheme is investigated to improve relay coordination and reconfigurations by agent communication for protecting a distribution network with DG embedded. Meanwhile, the use of a broker agent is aimed to provide a more dynamic and flexible approach for the anti-islanding protection of DG. This thesis also focuses on the development of an agent-based substation information management system that supports monitoring relay protection status, transformer condition assessment and information gathering in a substation and its associated distribution networks.

1.6 Contributions of Research

The main contribution of this thesis is to investigate the use of MAS in the context of power system automation, including distribution network protection and substation information management. The main aims are to determine the applicability of existing computer and information systems techniques to this domain, and to examine how different methodologies, such as multi-agent systems, can be combined into an integrated automation system. This thesis describes, for the first time, the original work on the development of the relaying schemes for the improvement of relay coordination using MAS. It is also the first time that an agent brokering mechanism is utilised to support communications between the relay and DG units for anti-islanding protection of DG. The major contributions arising from this thesis are outlined as follows.

- A multi-agent framework has been developed for distribution network protection and substation information management [65] [66]. A number of software agents have been designed based on a generic structure that defines a reasoning engine and three specific units to support agent decision making and operations. These generic agents can be easily reconfigured if the situated environments are changed. By using the generic agents to represent components of a power automation system, it is possible for the framework to more closely match the distributed nature of the system, and the flexibility and autonomy of the system are increased by allowing components to be added and removed at runtime. Within this framework, an agent communication infrastructure combining the FIPA communication protocols and the UDP protocol has been developed, which supports fast and reliable communications among the agents. In particular, an agent brokering mechanism has been proposed for optimising agent communications in a multi-agent system [67]. Simulation results indicate using the agent brokering mechanism, communication time consumption can be reduced in a large scale multi-agent system.

- Based on this multi-agent framework, a protective relaying scheme has been developed to improve relay coordination for the protection of distribution systems with DG integrated [68]. Using this scheme, a protection relay can be informed by the DG unit which is connected to or disconnected from its adjacent network in real-time. In this case, network topology changes due to the integration of the DG unit can be perceived by the employed generic agents, therefore the relay measurements and protection settings can be regulated in accordance with the preset control principles. Moreover, the performance of the proposed scheme in different protection issues have been investigated and simulation results suggest it is possible to avoid relay false tripping caused by the interconnection of DG units.

- An agent brokering-based scheme has been investigated within the proposed multi-agent framework for anti-islanding protection of the integrated DG units [69]. This scheme aims at transferring relay operating signals, *e.g.*, trip signals or reclosure signals, to the downstream DG units for coordinating their operations. In order to ensure the signals can be received by the downstream DG units in a short time frame, the proposed agent brokering mechanism has been applied. A broker agent that maintains the knowledge of network topologies and services provided by each agent is able to forward the requests from the relays to all of the relevant DG units. Particularly, a multi-brokering mechanism has been proposed to decrease the response time of this scheme in handling the situation that a large number of relays are tripped at the same time. Furthermore, different protection issues have been simulated and the results suggest the most valuable point of the proposed scheme is that it provides a flexible and scalable approach for anti-islanding protection of DG.

- A substation information management system has been designed based on the proposed multi-agent framework. A number of modules have been developed within this system to manage information acquired from different parts of a substation and its associated distribution network. The implementation of the system in substation asset management [70] has been investigated. Three specific issues, such as relay protection status monitoring, transformer condition assessment [71] and information gathering [72], have been performed.

1.7 Thesis Outline

This thesis is structured as follows:

Chapter 2 presents the proposed multi-agent framework in detail. The functions and specifications of a number of software agents derived from a generic structure are introduced, which are capable of control, communication, database and document management, analysis and display. Moreover, the developed agent brokering mechanism and the specifications of a broker agent are presented. Simulation studies on the evaluating the performance of the broker agent in terms of response time in handling requests are discussed.

Chapter 3 proposes a protective relaying scheme developed within the proposed multi-agent framework for the protection of a distribution network with DG integrated. The impacts of the integration of DG on protecting the power network and current solutions are discussed in detail. The development of this scheme are introduced, including two agent-based modules, *i.e.*, a relay module and a DG module. A number of simulation scenarios, concerning different fault types, *e.g.* phase-to-ground fault and phase-to-phase fault, different protection issues, *e.g.* overcurrent protection and impedance protection, as well as different situations *e.g.* a single fault issue, a multiple faults issue and an issue of interconnecting DG units are under investigation. Furthermore, the timing performance of the scheme is evaluated. The simulation results indicate the merits of the proposed scheme are in the flexibility, scalability and dynamic response for the protection of distribution systems.

Chapter 4 presents a novel agent brokering-based anti-islanding protection scheme developed based on the proposed multi-agent framework. The problems caused by islanded DG units and the related solutions are introduced followed by a description of the development of this scheme. Three modules, *i.e.*, a relay module, a DG module and a broker module, are presented. Two simulation scenarios are carried out, including multi faults occurrence and the change of network topologies, for investigating the performance of the proposed scheme in the coordination of relay and DG operations. Moreover, a series of experiments is carried out for the investigation of the time consumption of agent communications in the proposed scheme. A multi-brokering mechanism is developed for reducing the response time of the broker module in handling a large number of requests. The simulation results suggest the most valuable point of the proposed scheme is that it provides a flexible and scalable approach for anti-islanding protection of DG and the response time in tripping the DG units from the islanded condition is acceptable if there is not a large number relays being tripped at the same time.

Chapter 5 introduces the development of a substation information management system based on the proposed multi-agent framework. A system architecture and related technology, e.g. association rule mining (ARM)-based dissolved gas analysis (DGA) method, for transformer fault diagnosis are introduced. Four employed modules, *i.e.*, a transformer module, a device module, an information aggregation module and a user interaction module, consisting of a number of generic agents are then described. Furthermore, the implementation of this system in substation asset management is presented. A designed substation network model with its associated distribution network is introduced followed by descriptions of three specific applications, *i.e.*, monitoring relay protection operations, transformer condition assessment and information retrieval. Finally, messages transferred among the agents employed in the different modules and the tasks and communication protocols carried out by each agent are described.

Chapter 6 concludes the thesis based on the outcomes obtained in this study, followed by the discussion of the challenges of this work. Suggestions for future work are also listed in this chapter.

Chapter 2

A Multi-agent Framework for Power System Automation

This chapter describes in detail a multi-agent software framework for the protection of distribution network and substation information management. This framework is developed based on the e-Automation architecture that is firstly proposed at the University of Liverpool for power system control, condition monitoring and information management using the MAS technology [73]. Within this framework, a number of software agents designed based on a generic structure are developed for control, communication, database and document management, analysis and display. Furthermore, a middleware agent, known as a broker agent, is proposed and applied to improve the coordination among the agents employed in the proposed framework. The performance of a broker agent in terms of timing in communication is also evaluated.

2.1 e-Automation architecture

The e-Automation architecture defines a new generation of automation systems using the latest networking and agent technologies for information management, condition monitoring, and real-time control of a wide range of distributed industrial systems [74]. The e-Automation architecture contains a real-time simulator, a range of hardware consisting of microprocessors and embedded systems and data acquisition devices, real-time automation platforms, comprehensive software development systems and three IP networks, including a wireless local area network, which are used to carry out research in the area of network-based industrial automation.

In this architecture, a number of software agents were designed for undertaking different tasks. For example, a database agent is capable of providing the FIPA ACL-based access to a database. A document agent is responsible for the management of documents stored in a particular location, such as a directory of a filesystem. An ontology agent is used to implement the FIPA Ontology Service and register a number of predicates with the DF. The management of a specific data acquisition device is undertaken by a device agent. A plant agent is responsible for the monitoring and control of a single item of plant and a personal assistance agent is implemented using Component Object Model (COM) techniques for substation information management. Particularly, a mobile agent was proposed and evaluated for data analysis and remote control to improve the agent performance in applications consisting of multiple interactions or large data transfer over a low bandwidth or high latency network.

Comparing to the existing technologies for power system automation introduced in Section 1.1, there are a number of advantages provided by the e-Automation architecture, which are summarised as follows:

- **Flexibility:** The use of a multi-agent system enhances the flexibility of the system by permitting new devices and items of a plant to be added without changing software settings of the rest of the system. Although this advantage is reduced in the power systems domain by the fact that the system does not change rapidly, it would be useful to add a feature enabling the agent to read updated configuration rules at runtime.

- **Inherent distribution:** The e-Automation architecture that provides autonomy to its constituent components is well-suited to the inherently distributed nature of the power system. For example, the use of agents to represent objects such as transformers and circuit breakers is a natural fit to the system being controlled.

- **Integration:** Using a multi-agent system provides a convenient framework to represent different tasks and to integrate a variety of data sources. Rather than a number of separate software programs, all tasks are performed through the multi-agent system, enabling data to be shared between tasks. The use of a standard agent communication language provides a fixed communications mechanism which can be used by heterogeneous agents.

However, there are also several disadvantages of the e-Automation architecture in the applications of power systems:

- **Management of a large agent society:** Agent service request in the e-Automation architecture is based on DF, which provides a yellow pages directory service to agents. Each requester agent needs to search the service provider agents from a DF and contact with them directly. Furthermore, the DF does not guarantee the validity of the information provided in response to a search request. Therefore, the complexity of agent communication may be increased dramatically in a system consisting of a large number of agents.

- **Difficulty of integration with devices:** Although the device agents proposed in the e-Automation architecture provide a convenient interface to other agents, the implementation of a device agent is still performed in a similar way to that which would be used for a component of a traditional industrial automation system, and it is necessary to develop a specific device agent for each model of device. Therefore, the e-Automation architecture may not accommodate the latest development in this area.

2.2 A Multi-agent Framework

2.2.1 Improvements from the e-Automation architecture

A new multi-agent framework that extends the e-Automation architecture introduced in the previous section was developed and applied for the protection of distribution networks and substation information management. A JADE agent platform was built within this framework and six types of generic agents were developed based on this platform, *i.e.*, a database agent a control agent, a communication agent, an analysis agent, an interface agent and a document agent. Each agent provides one or more common capabilities, which can be specified and implemented to accomplish different tasks. Comparing with the e-Automation architecture, the improvements of this framework are highlighted:

- A generic structure of agents utilised in the e-Automation architecture was improved and a number of new types of software agents were designed based on this structure, such as the control agent, the communication agent, the analysis agent and the interface agent, for undertaking different tasks in the proposed framework.

- Agent coordination in a large scale multi-agent system was improved by the development of a middleware agent, known as a broker agent, to support communications between requester agents and service provider agents. The timing performance of the broker agent was also evaluated and compared with that of the conventional agents.

- In this framework, a module composed of agents with different functions was integrated with power devices for real-time communications and re-configurations of the power devices. Since the agents were developed based on the generic structure and provided one or more generic functions, the module could be easily embedded in different models of power devices. Therefore, the capability of integrating agents with power devices is significantly improved.

- The proposed framework was attempted to be applied in many new areas of power system automation, for example, distribution network protection, transformer fault diagnosis and condition monitoring, substation asset management, *etc.*

2.2.2 Functionality

As discussed in Chapter 1, a power automation system is to perform a variety of tasks, which operate at different timescales, have many characteristics and involve the exchanges of various types of data. The proposed multi-agent framework was developed based on a physical decomposition process that is commonly used as an approach for designing multi-agent systems [75]. In a multi-agent system, each object can be represented by an agent and the architecture which presents the modelled physical system can be derived by using this process and is easy to understand. Moreover, the physical decomposition can enhance the abilities of a system to cope with the changes. For instance, if an item is emerged into or removed from the the system, it is only necessary to inform the agents associated with this item of this change. In this case, all the functional agents which interact with those agents will be alerted. Furthermore, the extensibility of the framework can be improved, because some of the components employed in the framework can perform more than one role in different systems. Particularly, the functions provided by this multi-agent framework are listed as follows.

- **Data acquisition and storage:** takes raw data from sensors, translates it into numerical values and manages the historical data in databases.
- **Condition monitoring and fault diagnosis:** focuses on modelling transformer thermal dynamics and diagnosing the potential faults of a power transformer, which consists of data processing, intelligent classifier, rule-based reasoning, alarming and reporting.
- **Operation coordination of power devices:** supports agent-based relaying schemes for the protection of power networks with distributed

generations integrated.

- **Information management:** provides the more accurate and efficient information management and retrieval facility for documents related to the power system, including maintenance records for substation, technical documentations, *etc.*

- **User system interaction:** involves taking commands and queries from a user, and translating them into appropriate formats for submitting to other components of the system.

2.2.3 Agent platform

In the proposed framework, a JADE agent platform[76] designed based on the FIPA specifications [40] was established. The JADE platform simplifies the implementation of interoperable intelligent multi-agent systems through a middleware that complies with the FIPA specifications [77]. In the JADE platform, an individual agent can be defined as a single thread in a globally unique agent ID (AID). Behaviour abstraction is used to model the multiple simultaneous tasks performed by an agent. The use of a DF permits agents to be added and removed at runtime, as an agent providing a service may be substituted with another agent providing the same service. A directory entry for an agent includes (among other items) the agent name, service name, service type, ontologies (data models), protocols and a set of properties, which may be defined by the user, describing that service.

In addition, communication among the agents in the JADE platform adheres to the FIPA specifications, using the FIPA-specified ACL [78], which is a high-level agent communication language based on the speech acts and used as a standard message language by setting out the encoding, semantics and pragmatics of a message. An ACL message consists of an outer message structure, providing information expressed in some language understandable to both the sending and receiving agents [79]. Particularly, an ACL message is formatted in the following fields [78]:

- *sender*: the sender of a message;
- *receivers*: the list of receivers;
- *performative*: the communicative intention indicating what the sender intends to achieve by sending out the message, which can be `REQUEST`, if the sender wants the receiver to perform an action, `INFORM`, if the sender wants the receiver to be aware a fact, `QUERY_IF`, if the sender wants to know whether or not a given condition holds, and `CFP` (call for proposal), `PROPOSE, ACCEPT_PROPOSAL, REJECT_PROPOSAL`, if the sender and receiver are engaged in a negotiation, *etc.*;
- *content*: the actual information included in the message, for example, the action to be performed in a `REQUEST` message, the fact that the sender wants to disclose in an `INFORM` message;
- *language*: the content language, such as the syntax used to express the content that both the sender and the receiver must be able to encode/parse expressions compliant to this syntax to make sure the communication is effective;
- *ontology*: the vocabulary of the symbols used in the content and their meaning that both the sender and the receiver must ascribe the same meaning to symbols for the communication to be effective; and
- some fields used to control several concurrent conversations and to specify timeouts for receiving a reply such as `conversation-id, reply-with, in-reply-to, reply-by`.

The language used for inter-agent communications in the proposed framework is the FIPA SL language [80]. This syntax and its associated semantics are suggested as a candidate content language for use in conjunction with FIPA ACL. A FIPA SL content expression may be used as the content of an ACL message, which can be listed as follows:

- *proposition*, which may be assigned a truth value in a given context and used in the inform Communicative Act (CA) and other CAs derived from it;
- *action*, which can be performed as a content expression when the act is a request operator and other CAs derived from it;
- *identifying reference expression (IRE)*, which identifies an object in the application and can be used in the `inform-ref` macro act and other CAs derived from it.

2.3 Generic Agent Components

In the proposed multi-agent framework, a number of software agents are developed and executed on the JADE platform introduced in the previous section. As shown in Figures 2.1 and 2.2, all the agents are generated from the same basic structure and control loop which is derived based on FIPA agent development standards introduced in Section 1.2.3.

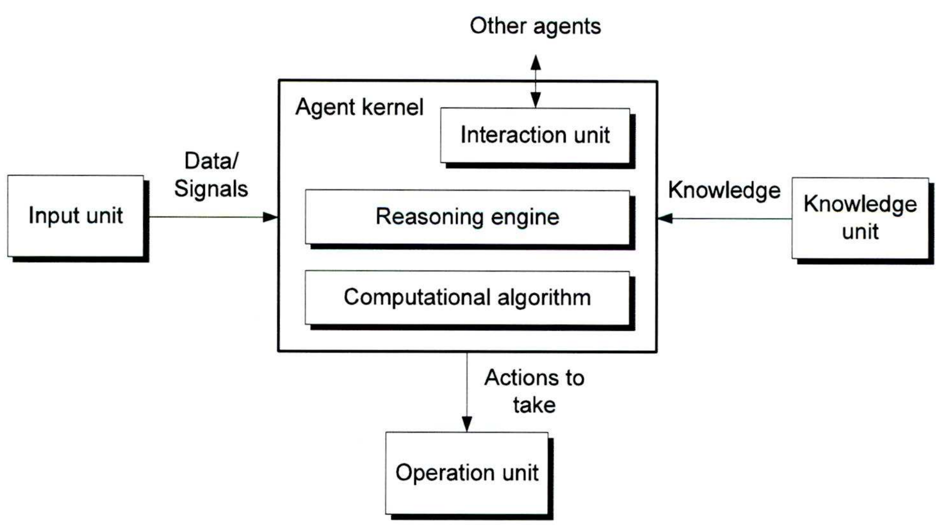


Figure 2.1: A generic agent structure.

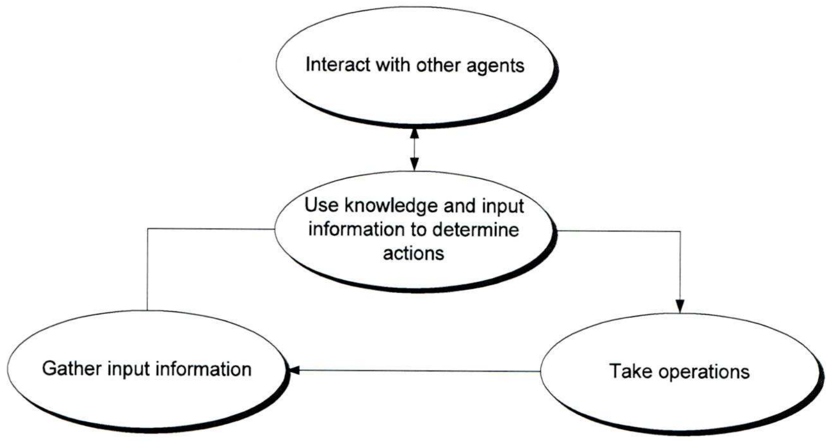


Figure 2.2: Agent control loop.

Each agent consists of an agent kernel and three units including a knowledge unit, an input unit and an operation unit. The agent kernel is composed of an interaction unit, which allows the agent to interact with other agents, and combined reasoning engines or computational algorithms to determine the actions to be carried out according to the input. The knowledge unit contains agent beliefs and goals for decision making. The input unit allows the agent to perceive its environment and receive data and signals from other agents or devices. The operation unit enables the agent to perform actions and operations. In this section, the detailed specifications of each generic agent are stated.

2.3.1 Database agent

Description

Figure 2.3 illustrates the structure of a database agent. The database agent provides FIPA ACL-based access to a database. It is able to insert and retrieve information to and from the database and to convert this information to agent understandable FIPA ACL messages. This conversion is performed by the reasoning engine built in the agent kernel using mapping rules stored in the knowledge base of the agent. The knowledge of the database agent includes database configuration rules (Java Database Connectivity (JDBC) driver, username, password) and transformation rules from database schema to global schema. The interaction unit allows it to receive queries and reply results.

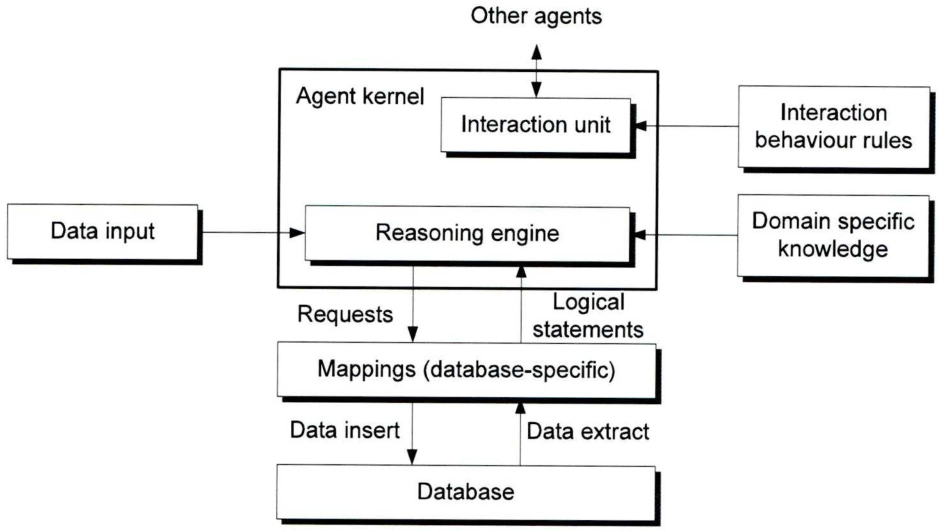


Figure 2.3: Structure of a database agent.

Agent specification

This specification defines the knowledge, inputs, operations and interaction protocols required by the database agent. Using this information, it is possible to modify a generic agent so that it can act as a database agent.

- Knowledge
  - Transformation rules from database schema to global schema.
  - Database configuration rules(Java Database Connectivity (JDBC) driver, username, password).
  - Interaction behaviour rules (message sender, receiver, performative, contents).
- Inputs

  - Data received from sensors or other agents.
  - Operational signals received from power devices.

- Operations

  - Append/Insert data to a database.
  - Query historical data from a database.

- Interaction protocols

  - FIPA query (responder) - used by other agents to query the database.
  - FIPA subscribe (initiator) - allow it to establish a subscription with a data provider agent (*e.g.* a control agent or an analysis agent) so that new events are transferred into the database as they occur.

2.3.2 Control agent

Description

The structure of a control agent is shown in Figure 2.4. The control agent connects to a power device (*e.g.* a protection relay, a circuit breaker and a DG unit) for the management of device configurations, inputs and outputs. It maintains up-to-date knowledge of device status and collects the operational signals from the input unit. The interaction behaviour rules enable it to receive and send requests from and to other agents. Furthermore, based on current status of the connected device, received requests from other agents and the specified control principles, the reasoning engine is able to take actions and sends commands to the device via the operation unit.

Agent specification

- Knowledge

  - Type of the device.
  - Current status of the device.

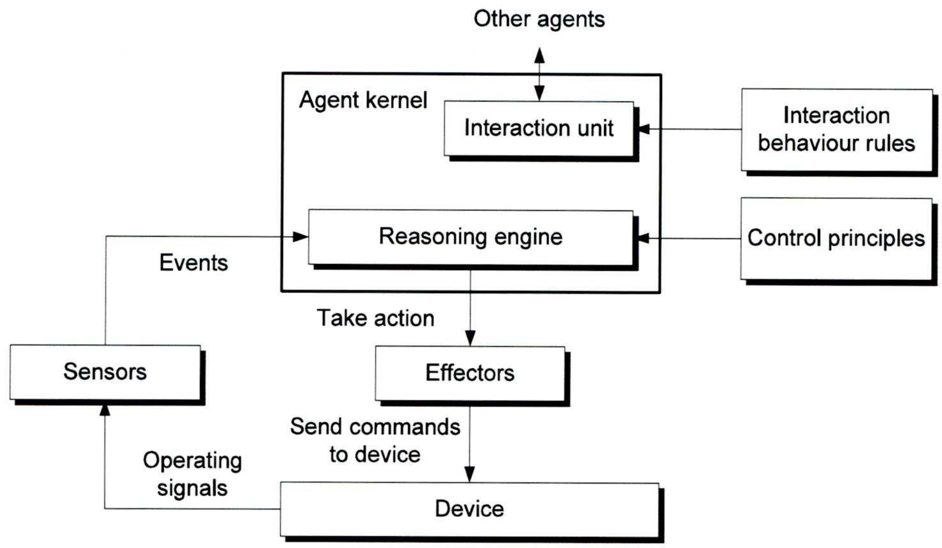


Figure 2.4: Structure of a control agent.

  - Device configurations.
  - Control principles.
- Inputs
  - Operational signals received from the device.
- Operations
  - Receive operational signals from the device.
  - Send commands to the device.
- Interaction protocols
  - FIPA query - used by other agents to retrieve data from the device.
  - FIPA subscribe - used by other agents to establish a subscription to be notified whenever the value of a channel changes.
  - FIPA request - used to reconfigure the device.

2.3.3 Communication agent

Description

The communication agent, shown in Figure 2.5, is responsible for message exchange in the proposed framework. It maintains the knowledge of the communication protocols, agent coordination rules and ACL message templates. The communication protocols are based on the combined FIPA and UDP protocols, because the use of FIPA specified platform provides a set of standard services (*e.g.* AMS and DF) to perform agent lifecycle management, communications and service discovery and UDP-based protocols support real-time applications. According to the requests from other agents, an ACL message is generated by the reasoning engine using the pre-defined message templates and sent over to target agents.

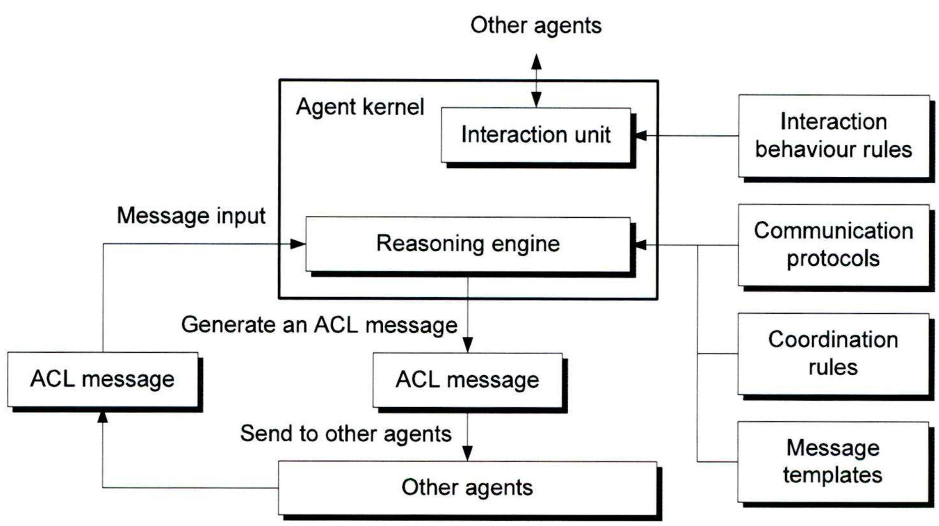


Figure 2.5: Structure of a communication agent.

Agent specification

- Knowledge
  - Communication protocols.

  - Coordination rules.
  - ACL message templates.

- Inputs
  - ACL messages.

- Operations
  - Generate ACL messages.
  - Send the messages to other agents

- Communication protocols
  - FIPA request (initiator) - used to forward requests to other agents.
  - FIPA request (responder) - used to receive messages from other agents.
  - UDP protocols - used for real-time agent communication.

2.3.4 Analysis agent

Description

Figure 2.6 illustrates the structure of an analysis agent. The analysis agent is capable of processing data in the proposed multi-agent framework. Depending on the requirements received from other agents, such as data types, amount, time period, *etc.*, it requests a database agent to retrieve the related data from a database and receives the data through the data input unit. The computational algorithms integrated within the reasoning engine of the analysis agent are responsible for the calculation of the received data and the results can be returned to the requester agents.

Agent specification

- Knowledge

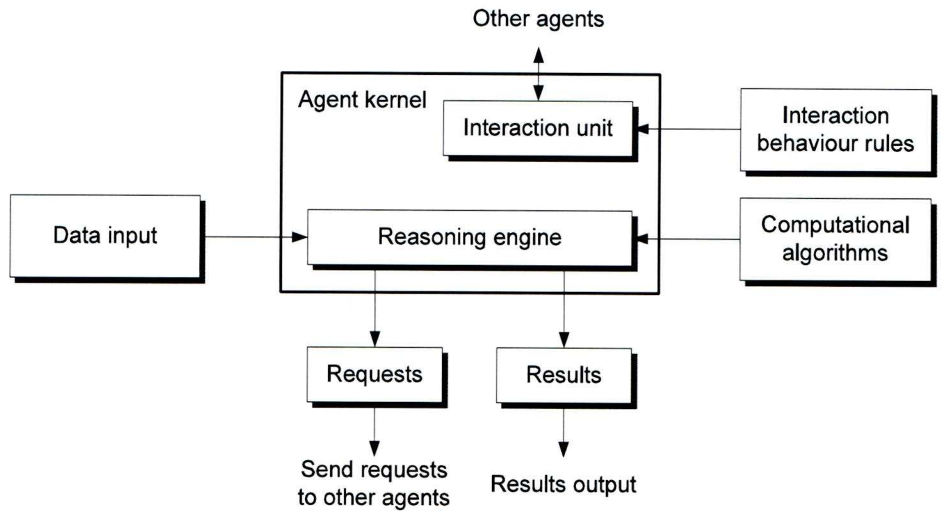


Figure 2.6: Structure of an analysis agent.

  - Data calculation algorithms.
  - Identification number of database agents.

- Inputs
  - Data retrieved from a database.
- Operations
  - Request a database agent for data retrieval.
  - Return analysis results.
- Interaction protocols
  - FIPA request (initiator) - used to forward requests to other agents.
  - FIPA request (responder) - used to receive messages from other agents.

2.3.5 Interface agent

Description

The interface agent, shown in Figure 2.7, in the proposed framework provides a link between the agent community and users via a human machine interface (HMI). The knowledge can be transferred from the representation used by the multi-agent system into that used by the HMI. It is also capable of carrying out tasks by using the directory services provided by the agent platform to locate appropriate agents to perform these tasks.

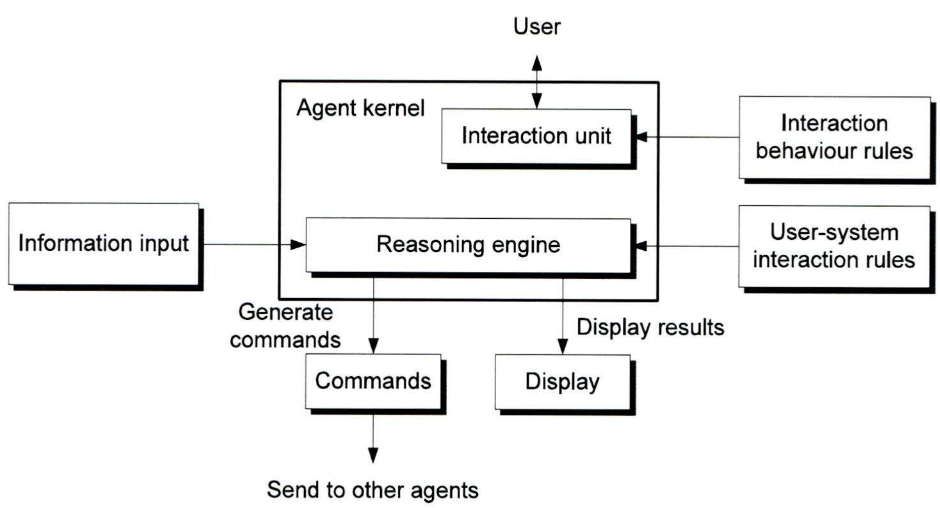


Figure 2.7: Structure of an interface agent.

Agent specification

- Knowledge
  - A representation of the current state of the system, obtained at runtime from other agents.
  - A set of user-system interaction rules required to convert from the global system ontology to a representation used by the HMI.
- Inputs

  - Information input from graphical user interface.

- Operations
  - Generate configuration files for data analysis and control agents, and launch agents.
  - Display information on graphical user interface.

- Interaction protocols
  - FIPA subscribe (initiator).
  - FIPA request (initiator).
  - FIPA query (initiator).

2.3.6 Document agent

Description

The document agent is responsible for the management of documents stored in a particular location, such as a directory of a filesystem. The agent has two main responsibilities: to ensure that it stores up-to-date statistics regarding its document collection, and to carry out queries on behalf of other agents. The structure of the document agent is shown in Figure 2.8, which is similar to that of the database agent. However, the document agent requires a sensor to notify it when new documents are added to the repository and generate metadata for use by an information retrieval algorithm. This algorithm can then be used to determine the relevance of a document to a particular query.

The reasoning engine used by the agent is relatively simple. It passes the query received from another agent to the document ranking algorithm, which generates a set of matching documents. These are then returned to the querying agents. Furthermore, the document agent can be used to retrieve the full text of a document.

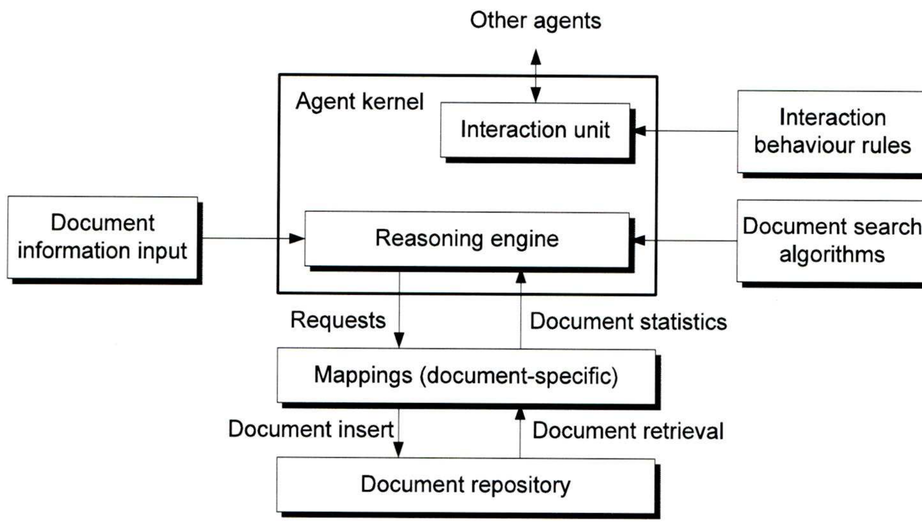


Figure 2.8: Structure of a document agent.

Agent specification

The knowledge, inputs and sensors of the document agent allow it to generate document statistics and retrieve documents relevant to a query. In order to allow other agents to query for relevant documents, the FIPA Query protocol is used. The FIPA Request protocol allows other agents to request the text of a document.

- Knowledge
  - Document collection statistics.
  - Document ranking algorithms.
- Inputs
  - A specified location where the documents are stored.
  - Document information, for example, index or key words.
- Operations
  - Generate the required document statistics for the particular retrieval methodology.

  - Retrieve the list of documents relevant to a query.
  - Retrieve document full text on request.
- Interaction protocols
  - FIPA Query
  - FIPA Request

2.4 Agent Communication and Protocols

2.4.1 Communication requirements in power system

Broadband data communications are popularly in use for electric utilities. In particular, the communication requirements for power systems are classified as real-time operational communication, administrative operational communication and administrative communication.

Real-time operational communication encompasses communication in real time that is required to maintain operation of the power system, which can be divided into real-time operational data communication and real-time operational speech communication. Real-time operational data communication is characterised by the fact that interaction must take place in real time with hard time requirements, encompassing tele-protection and power system control (PSC). For tele-protection purposes, messages should be transmitted within a very short time frame. Maximum allowed time is within the range of 12-20 ms, depending on the type of protection scheme. The requirement has its origin in the fact that fault current disconnection shall function within approximately 100 ms. Moreover, PSC mainly includes supervisory control of the power process on secondary or higher levels. These systems can be categorised as SCADA/energy management system (EMS).

In addition to real-time operational communication, administrative operational communication is characterised by those interactions which do not need to take place in real time, including asset management, fault location, metering and transfer of settlement information, security system, *etc.*

Furthermore, administrative communication includes voice communication and facsimile within the company (also between the offices that are at different geographical locations) as well as to/from the company where the communication has an administrative purpose. It contains telephony, facsimile (over PSTN and/or cellular network) and e-mail which serves as substitute and/or supplement to internal and external regular postal services.

2.4.2 FIPA-compliant agent communication

As introduced in Section 2.2.3, agent communications in the proposed multi-agent framework are based on a standard FIPA platform - JADE, which provides a set of libraries in some programming language or languages (usually Java) which simplify the task of the agent programmer. Agent communication protocols supported by the agent platform specifies the rules of interaction governing a dialogue between agents.

The main advantage of using such a platform is the reduction of the amount of implementation effort required to implement the architecture through the use of an off-the-shelf agent platform. Also, the platform would be able to interoperate with other FIPA systems if that was required. JADE architecture is based on a modular structure, in which a platform splits into a number of containers, which may run on different machines, and communicate via Java RMI. Communication between agents on the same container uses Java method calls. The utilised FIPA-compliant agent interaction protocols are listed as follows:

- *FIPA Agent Message Transport Protocol (MTP)*: The MTP is based on the transfer of data representing the entire agent message including the message envelope in an HTTP request. The HTTP data transfer is a two-step process: the sender makes an HTTP request and once receiving the data the receiver sends out an HTTP response. The receiver then parses the message envelope and the message is handled according to the instructions and information given in the message envelope.

- *FIPA Query Interaction Protocol*: This protocol allows one agent to request to perform some kind of action on another agent.
- *FIPA Request Interaction Protocol*: This protocol allows one agent to request another to perform some action.
- *FIPA Subscribe Interaction Protocol*: This protocol allows an agent to request a receiving agent to perform an action on subscription and subsequently when the referenced object changes.

2.4.3 UDP-based agent communication

Due to the requirements of real-time operational communication introduced in the previous section, messages should be transmitted among agents within a very short time frame for protecting the power system. Comparing with the Transmission Control Protocol (TCP) protocol that is responsible for reliable communication between two end processes, the main advantage of using the User Datagram Protocol (UDP) protocol is the extremely fast message transmission. In this case, the UDP protocol is used in the proposed framework for sending messages between agents which are required to work under timing constraints, such as device agents and communication agents. Each agent listens to messages on a UDP port and a broadcast port, which allows for both peer-to-peer and broadcast messaging. However, the drawbacks of UDP are that the UDP protocol does not provide reliable service. For example, a packet may not be delivered, or delivered twice, or delivered out of order and there is no indication of whether a packet is delivered or not.

2.4.4 Combined FIPA and UDP agent communication model

A possible solution to the real-time and limited device difficulties of a FIPA platform and the scalability and reliability problems of a UDP-based platform is to combine the two. This combined platform might have a similar conceptual architecture to that of the Lightweight and Extensible Agent Platform (LEAP) system [81]. It extends the JADE FIPA platform by providing a "container" based on the Java 2 Micro Edition, which hosts JADE agents and communicates with a host platform using a proprietary socket-based protocol. In this framework, the utilised JADE platform is extended to consist of a FIPA-based platform along with a gateway agent, which allows any number of independent agents to join the system. These agents are implemented in the framework as described for supporting the UDP protocol. The function of the gateway agent is to translate messages from the format used by the UDP-based agents to FIPA message objects as defined by the FIPA platform, and to pass them to the FIPA platform through Agent Communications Channel (ACC) for handling. This means that all platform services for the UDP-based platform can be provided by the FIPA platform, but the UDP-based agents retain the ability to communicate with each other in a peer-to-peer manner, without using the platform, when required.

2.5 Middleware Agent: A Broker Agent

One of the important contributions to the proposed multi-agent framework is that a middleware agent, also known as a broker agent, is developed for the improvement of agent coordination. In this section, the detailed descriptions of the broker agent, including the structure, agent brokering mechanism and its timing performance within the framework are given.

An agent employed in a multi-agent system is not totally isolated. It is essential for an agent to communicate, coordinate and cooperate with other agents whenever it is permitted. It is possible to hard-code knowledge of the agents or use a broadcast-based protocol, such as the Contract Net Protocol [82] if only a few agents are employed in a multi-agent system. However, if the number of agents increases, it is difficult for an agent to find other agents that provide the required services under this circumstance. Previously, this problem was addressed by the use of middleware agents, which identify both the preferences and capabilities of the participants, and routes both requests and replies appropriately. Examples of the use of the middleware agents include InfoSleuth [83] IMPACT [84] and RETSINA [85] [86].

Within the proposed multi-agent framework, a middleware agent, known as a *broker* agent, was developed. Each agent employed in the multi-agent system can register its services with the broker agent to assist a client in discovering and connecting with one or more relevant service providers. Particularly, when consider a scenario such as a client wishes to use the broker agent, instead of searching for a suitable service providing agent, it submits its request to the broker agent, which then handles the locating of suitable agents, and sends the request to those agents. In this case, the client agent does not have to communicates directly with the service providing agents.

In this section, the development and implementation of a broker agent in the proposed multi-agent framework is introduced, including the design of an agent-brokering mechanism, brokering agent interaction protocols and the evaluation of the communication timing performance of the broker agent in different situations.

2.5.1 Broker agent structure

Description

The broker agent, also designed based on the generic agent structure introduced in Section 2.3, is implemented in the framework for coordinating the operations of the agents registered on the agent platform. Specifically, it is able to receive requests from an agent, locate the target agents which provide the required services and forward these requests to them, and return the results back to the requester. Figure 2.9 illustrates a structure of the broker agent.

The knowledge unit of the broker agent is composed of agent brokering protocols that support broker agent communications, a knowledge base containing the up-to-date information of the service provider agents, and the pre-defined ACL message templates. Two types of messages are received as inputs of the

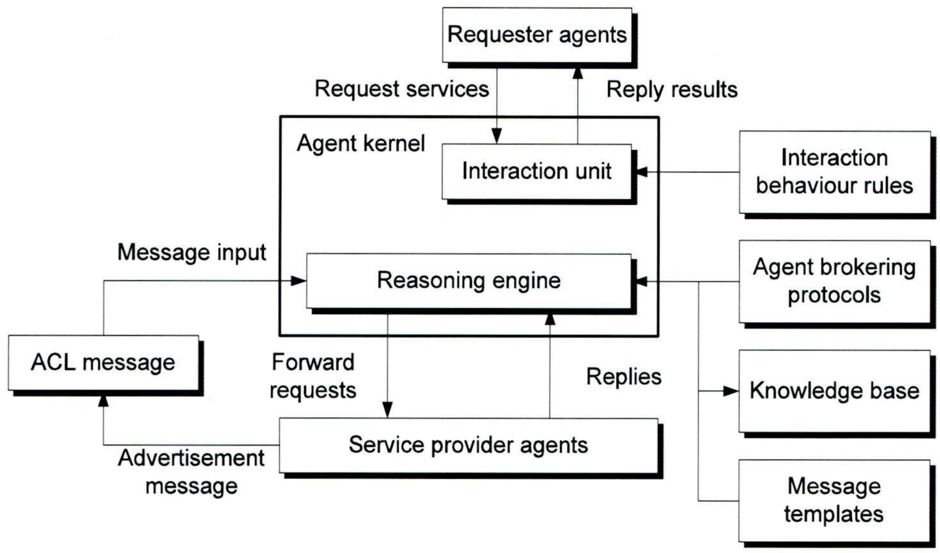


Figure 2.9: Structure of broker agent.

broker agent, *i.e.*, advertisement messages from the service provider agents and request messages from the requester agents. In the following subsection, the agent brokering mechanism and the communication patterns of the developed broker agent are introduced in detail.

Agent specification

- Knowledge
  - Agent brokering protocols.
  - Service information of the provider agent.
  - ACL message templates.
- Inputs
  - Request messages.
  - Advertisement messages.
- Operations
  - Match the target agents in accordance with the received requests.

  - Forward the received requests to the target agents.
  - Return the result messages back to the requesters.

- Interaction protocols
  - FIPA Query
  - FIPA Request
  - FIPA Brokering Interaction Protocol

2.5.2 Agent brokering mechanism

FIPA brokering interaction protocol

Apart from the FIPA agent development standards introduced in Section 1.2.3. FIPA also defines the interaction protocols for a broker agent, the FIPA Brokering Interaction Protocol (BIP) [87], which supports communications between the broker agent and other agents in the proposed framework. The FIPA BIP is a macro IP, since the proxy communicative act ([FIPA00037]) for brokerage embeds a communicative act as its argument, therefore, the IP for the embedded communicative act is also embedded in FIPA BIP. When the embedded communicative act includes some actions that would be done by the agents determined by broker agents, this IP would be extended for notifying the result of the actions. In Figure 2.10, a general description of FIPA BIP is given.

FIPA BIP defines that a broker agent should record some of the ACL parameters ([FIPA00061]), for example, `:conversation-id`, `:reply-with` and `:sender`, of the received proxy message to forward back the replying message to the corresponding original agent (the sender of the proxy message). In the proposed multi-agent framework, FIPA BIP was utilised as the standards for the design of an agent-brokering mechanism, which will be introduced in the following section.

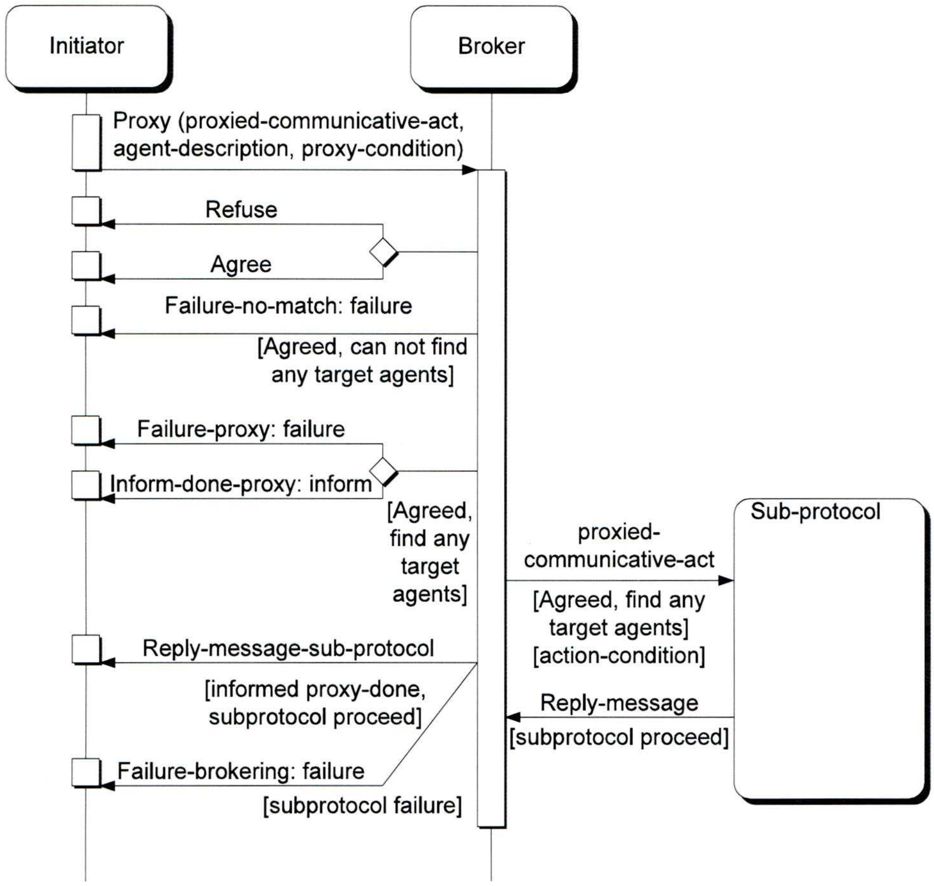


Figure 2.10: FIPA Brokering Interaction Protocol.

Broker agent communication pattern

Based on the FIPA BIP, an agent-brokering mechanism was designed in the proposed multi-agent framework to support agent coordination, which offers a set of communication facilitation services to the agents using the knowledge of the requirements and capabilities of those agents. Figure 2.11 displays the basic interaction patterns and the flowchart of the agent-brokering mechanism. Agents employed in the proposed framework are divided into two categories, the provider agents and the requester agents. A broker agent was designed for maintaining a knowledge base of advertised information of the provider agents and using this knowledge to match the agents that can provide the suitable services. A simple example is that a database agent may advertise its capabilities to the broker agent that it is able to request a database in SQL.

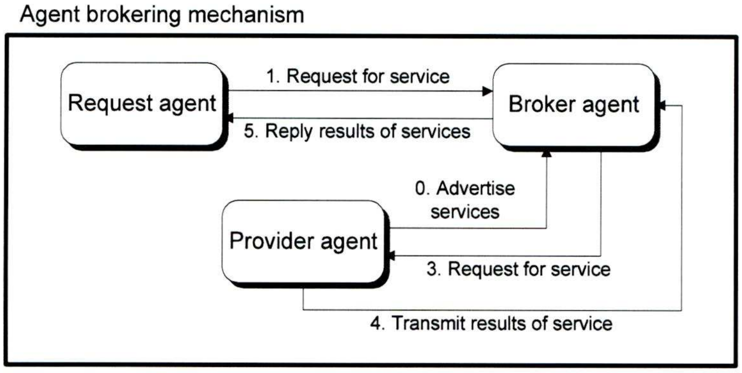


Figure 2.11: Interaction patterns of an agent-brokering mechanism.

As a result, the use of the agent-brokering mechanism can significantly simplify the task of agent interaction in a multi-agent system. Additionally, the broker agents also enable a system to be adaptable and robust in the dynamic situations, supporting scalability and security control. As shown in figure 2.11, the interaction pattern of the agent-brokering mechanism indicates that a requester agent is to ask a broker agent to find one or more agents that can answer the query. The broker agent then determines a set of appropriate provider agents to which to forward the query, sends the query to those agents and relays their answers back to the original requester.

2.5.3 Evaluation of the agent-brokering mechanism

In order to evaluate the performance of the proposed agent-brokering mechanism, a series of experiments has been undertaken, which focused on testing the response time of the broker agent to a request. Agents utilised in the experiments are executed on the proposed agent platform and simulated to run on its own processor. Agent communication is modelled to be carried by a network whose bandwidth is set to be 10 MB/s with a pre-set latency time of 100 ms per message. In this case, the total time consumption of one message delivery (T_{deliver}) is defined as:

$$T_{\text{deliver}} = T(msg, sender, receiver) + \delta(sender, receiver) \tag{2.5.1}$$

where, $T(msg, sender, receiver)$ is the time consumption of a message msg delivered from a *sender* to a *reciever* and the network latency between the sender and the receiver is presented by $\delta(sender, receiver)$. Therefore, for one request, the response time of the broker agent is presented as follows:

$$\begin{aligned} T_{\text{BA}} &= T_{\text{req}}(msg, \mathcal{A}_{\text{r1}}, \mathcal{A}_{\text{ba}}) + \delta(\mathcal{A}_{\text{r1}}, \mathcal{A}_{\text{ba}}) \\ &\quad T_{\text{req}}(msg, \mathcal{A}_{\text{ba}}, \mathcal{A}_{\text{p1}}) + \delta(\mathcal{A}_{\text{ba}}, \mathcal{A}_{\text{p1}}) \\ &\quad T_{\text{rep}}(msg, \mathcal{A}_{\text{p1}}, \mathcal{A}_{\text{ba}}) + \delta(\mathcal{A}_{\text{p1}}, \mathcal{A}_{\text{ba}}) \\ &\quad T_{\text{rep}}(msg, \mathcal{A}_{\text{ba}}, \mathcal{A}_{\text{r1}}) + \delta(\mathcal{A}_{\text{ba}}, \mathcal{A}_{\text{r1}}) \end{aligned} \tag{2.5.2}$$

where $\mathcal{A}_{\text{r1}}$, $\mathcal{A}_{\text{ba}}$, and $\mathcal{A}_{\text{p1}}$ are represented as the requester agent, the broker agent and the provider agent, respectively.

According to the equation 2.5.2, if there is one target agent ($\mathcal{A}_{\text{p1}}$) in the system and the requester agent ($\mathcal{A}_{\text{r1}}$) only sends out one request, the average response time of the broker agent measured in the experiment is approximately 490 ms, including the latency time (400 ms) for four time message transfers and agent communication time (90 ms). Specifically, 8 target agents are then executed in another experiment and the number of the queries performed by the request agent are varied from 5 to 30 in increments of 5 each time. Table 2.1 lists the response time of the broker agent in each query process.

As shown in Figure 2.12, the response time of the broker agent increases along with the increasing of the number of the queries and the target agents. Especially, when the number of queries is more than 25 and there are 6 targets to be contacted or the number of queries is more than 20 and there are 8 targets, the time consumption of the total brokering process significantly increases.

2.5.4 Comparison

In order to compare the time consumption of the proposed agent-brokering mechanism and the normal client-server method in agent communication, an

Table 2.1: Response time of a single requester (ms).

	Target agents			
Requests	2	4	6	8
5	603	634	666	712
10	743	790	865	916
15	868	947	1041	1134
20	994	1103	1278	1338
25	1134	1259	1400	1665
30	1275	1400	1712	1900

experiment was undertaken. In this experiment, the total number of the executed target agents is 10 and the number of the requester agents is increased from 4 to 20 by 4 each time. Moreover, each requester agent is capable of sending 20 request messages to the target agents. Table 2.2 lists the time consumptions of both of the methods in this experiment, where BA is defined as the proposed broker agent method, and CS expresses the client-server method.

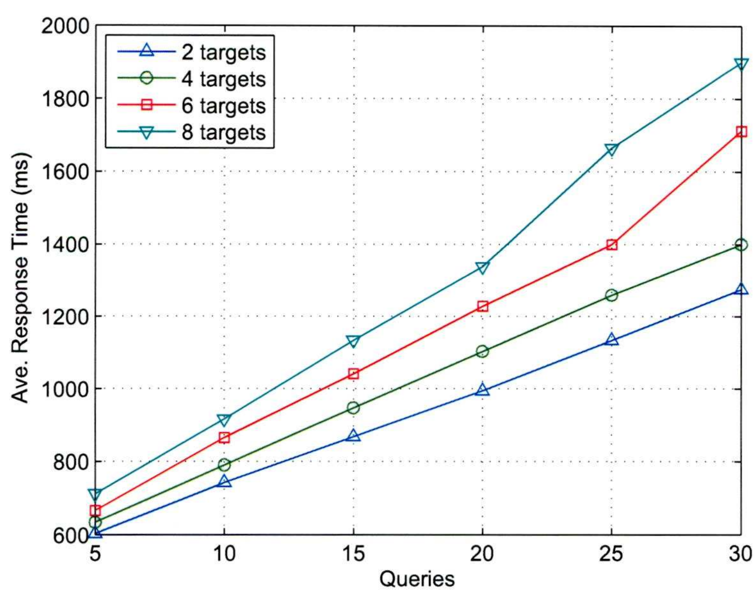


Figure 2.12: Response time of single requester.

Table 2.2: The time consumption of the broker agent method and the client server method (s).

	Target agents				
Method	4	8	12	16	20
BA	4.15	7.79	11.38	15.13	18.53
CS	2.15	6.28	12.93	18.66	25.71

Figure 2.13: The comparison of the broker agent method and client server method.

Arising from the table, when the number of target agents is smaller, the performance of the client-server method is better than that of the broker method in agent querying. However, if the number of target agents increases, the proposed agent-brokering mechanism uses less time in agent requesting than the normal client-server method, which is indicated in Figure 2.13. In this case, the proposed agent-brokering mechanism can provide a more efficient agent communication method in the proposed multi-agent framework for the development of agent-based protection, fault diagnosis and asset management.

2.6 Summary

This chapter has introduced a multi-agent framework developed based on the e-Automation architecture, for protection of distribution networks and substation asset management. Comparing with the e-Automation architecture, this framework represents four main improvements, such as the completed generic agent structure, the development of a broker agent for optimising agent coordination, the enhanced capability for integrating agents with power devices and the applications in many new area of power system automation.

In addition, the agent platform established in this framework based on the JADE platform for the execution of agents and management of message delivery was described. Six types of generic agent components and their specifications were introduced in detail. Particularly, the broker agent proposed in this framework was presented, including the broker agent structure, the agent brokering mechanism and protocols and the evaluations of timing performance in agent coordination.

Chapter 3

An Agent-based Relaying Scheme for Protection of Distribution Networks

This chapter introduces an agent-based relaying scheme for the protection of power networks with distributed generation (DG) integrated. Traditionally, a protection relay uses runtime measurements and characterised settings as the basis of fault determination and operation. Two or more relays, installed in series, are considered to be coordinated, providing a specified operating sequence [88]. Most of the existing relaying schemes are based on pre-defined procedures that are set case-by-case with the fixed parameter settings and determined in manufactory. Moreover, those relaying schemes work based on the assumption that the topologies of the power networks are unchanged and the transmission power varies only within a small boundary to ensure relaying accuracy.

However, the integration of DG raises many safety and technical problems, which will affect the amplitude, direction and duration of fault currents, reverse power flow and voltage profile, and also reduce the reach area of an impedance relay [89] [90]. Furthermore, it will cause problems when islanding occurs as the consequence of a fault in the network, since DG may continue its operations after the utility supply has been disconnected [91]. The existing relaying schemes face over increasing challenges.

Based on the multi-agent framework described in Chapter 2, a new option for protecting DG integrated distribution networks has been investigated. In this chapter, the problems and impacts of the integration of DG on protecting distribution networks are analysed. A structure of the proposed agent-based relaying scheme is then presented, followed by descriptions of the functions and the integrated algorithms of each agent. Furthermore, simulation studies for evaluating the performance of the proposed scheme in the different situations are introduced and the discussion of the simulation results is given.

3.1 Protection of Distribution Networks

3.1.1 Distributed generation

DG, also known as embedded generation, is electricity generation, which is connected to the distribution network rather than the high voltage transmission network [4]. It is typically a small generation such as renewable generation, including small hydro, wind and solar power, *etc.* As mentioned in [92], there is no standing international definition for DG, but generally distributed power sources have some characteristics in common:

- Compared to conventional power plants, the rating of DG is smaller.
- They are often owned privately.
- They are not centrally distributed.
- They are often connected to MV or LV distribution networks.
- They do not contribute to frequency or voltage control.
- Usually they are not considered when the local grid is planned, and therefore there are infrastructures needs such as means of communication.

The 2007 U.S. Department of Energy report [93] summarised that DG has the following potential benefits:

- increasing electric system reliability,
- reducing peak power requirements,
- improvement of power quality,
- reduction of land use impacts for transmission and distribution, and
- reducing vulnerability of the electric system to terrorism and providing infrastructure resilience.

3.1.2 Impacts of DG on distribution network protection

DG units composed of a number of distributed generators are interconnected at substations, distribution feeders or customer loads for providing additional power and capacities with a limited size (normally lower than 10 MW) [94]. However, the capabilities of a distributed generator varies from time to time, which significantly degrades the accuracy and reliability of relay operation. In this case, unacceptable relay operation, such as nuisance and sympathetic tripping due to the altered fault levels and unsynchronised reclosure caused islanding, may occur as DG is integrated with the networks. For example, as one of the common faults, a short circuit fault normally results in an overcurrent, which is significantly higher than the operational or nominal current. However, power flow on the distribution networks may be changed due to the variation of DG capacities and therefore the real fault current may not be detected by the protection relays. Consequently, a serious consequence may appear if an overcurrent relay does not trigger in sufficient time or the maximum fault distance that triggers an impedance relay in a certain impedance zone or in a certain time is reduced. Moreover, a faulted area, isolated by protection relays, will be islanded if DG is integrated and an arc is still charged, which may cause undefined voltage magnitudes or frequency changes [91].

In addition, the traditional distribution networks are planned as passive networks that power flows from high voltage levels down to loads located along with radial feeders [95]. The protection issues in such networks are simple since it is considered that the currents have only one direction (from high to low). In this section, the impacts of integrating DG on protection of the distribution networks are discussed, including the short circuit power and overcurrent protection and the reduced reach of impedance relays.

Short circuit power and overcurrent protection

The fault current level that describes the effect of faults in terms of current or power indicates the short circuit current or (apparent) power boost [4]. In [92], the fault level fl is defined as:

$$fl = i = \frac{1}{|\underline{z}_{th}|} \tag{3.1.1}$$

where i is the fault current related to the nominal current and $\underline{z}_{th}$ is the inner impedance of the Thevenin representation of the network in p.u. The typical fault levels in distribution networks are in a range of 10-15 p.u., where 1 p.u. corresponds to the rated current.

Phase-to-phase or phase-to-ground faults normally result in an overcurrent which is significantly higher than the operational or nominal current. The fault current has to be distinguishable from the normal operational current. To fulfill that, there has to be a powerful source providing a high fault current until the relay is tripped.

However, with DG integrated in the network, the fault impedance $\underline{z}_{th}$ decreases due to parallel circuits, therefore the fault level increases and unexpected high fault currents might occur in case of a failure. Moreover, another influence of DG on fault currents is that the fault current contribution from the generator may pass a relay in reverse direction. Consequently, the fault level will decrease and the real fault will be ignored by a relay. This situation puts components at risk since they were not designed to operate under that circumstance.

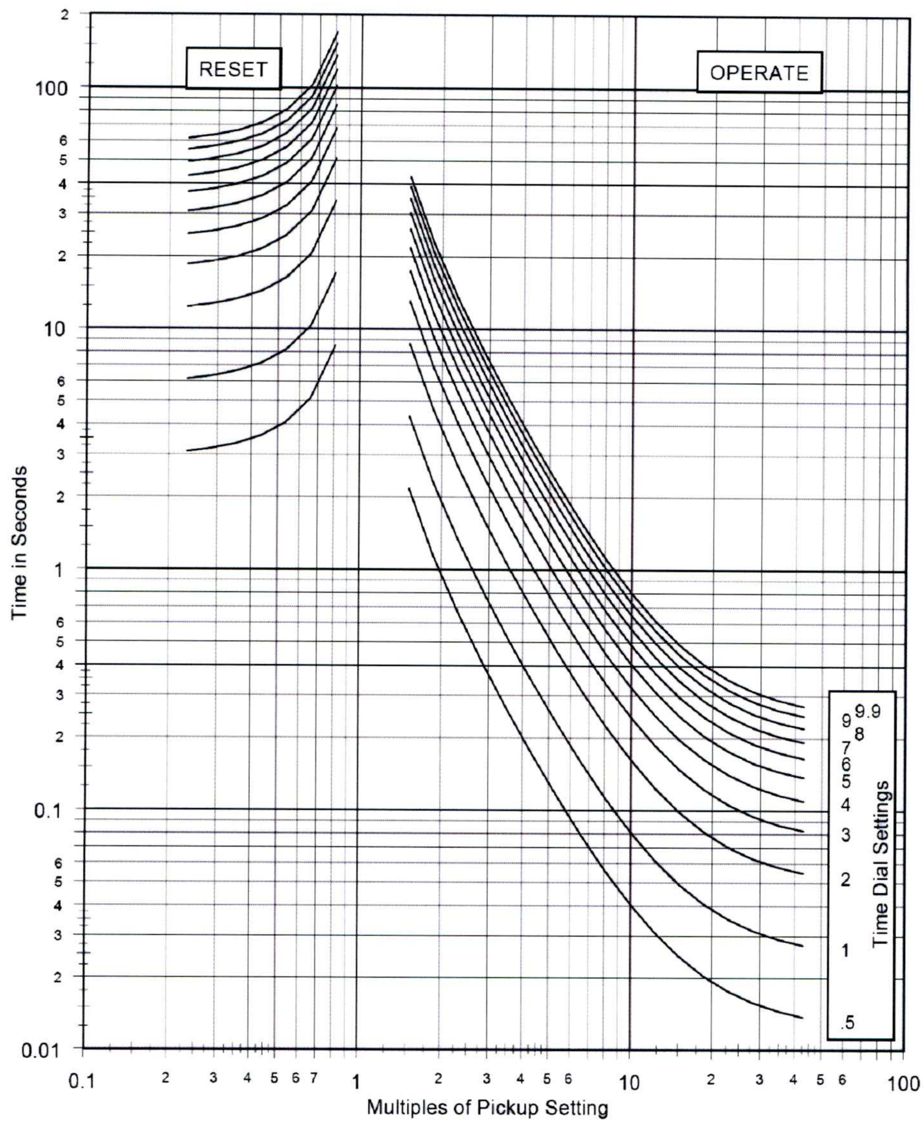


Figure 3.1: IEEE extremely inverse TOC curves.

Particularly, for a time-overcurrent relay which operates with a time delay, the higher current will result in a faster operation of the relay. Figure 3.1 shows the typical characteristics of an time-overcurrent relay following IEEE extremely inverse TOC curves [96]. According to the above discussions, if the fault level increases, the relay operating time will be shorter than its original setting, whereas the fault level decreases, the operating time of a relay will be longer. Therefore, incorrect relay operations may appear if the relay can not detect the actual current in the power network.

As shown in Figure 3.2, a radial power network with an embedded generator DG_1 connected to the busbar b_2 that supplies part of the local loads. An additional current (I_{dg1}) provided by DG_1 is inserted into the network, including a reverse current (I_{rvs}) and a forward current (I_{fwd}), where $I_{dg1} = I_{rvs} + I_{fwd}$. Assuming a short circuit with resistance at point F_1, if no DG unit is integrated, the current measurement of R_1 is:

$$I_{r1} = I_{f1} \tag{3.1.2}$$

where I_{r1} is the current detected by R_1 and I_{f1} is the fault current. The protection operating time of R_1 is T_{r1} in accordance with its characteristics.

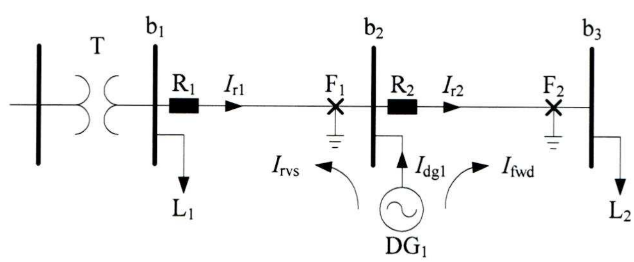


Figure 3.2: Short circuit power.

However, if a DG unit (DG_1) is connected to b_2, an additional reverse current I_{rvs} then flows from b_2 to b_1. In this case, if the same fault occurs at F_1, the measurement of R_1 is altered as:

$$I'_{r1} = I_{f1} - I_{rvs} \tag{3.1.3}$$

Evidently, in this case, the detected fault current value is lower than I_{r1}. Therefore, with the original relay operation settings, the operating time of R_1 becomes longer. That is,

$$T'_{r1} > T_{r1}$$

where T'_{r1} is the operating time of R_1 when DG_1 is connected to b_2. Consequently, R_1 may be operated later and false tripping may occur.

Similarly, assuming another short circuit fault with resistance at point F_2, as indicated in Figure 3.2, if no DG is integrated, the current measured by R_2 (I_{r2}) is:

$$I_{r2} = I_{f2} \tag{3.1.4}$$

where I_{f2} is the fault current. When DG_1 is connected to b_2, a forward current is involved into the adjacent downstream network. If the same fault appears at F_2, the current measured by R_2 is changed to:

$$I'_{r2} = I_{f2} + I_{fwd} \tag{3.1.5}$$

where I_{fwd} is the additional reverse current injected by DG_1. Obviously, the measurements of R_2 during the fault in the DG integrated network is lower than that without DG connection. Accordingly,

$$T'_{r2} < T_{r2}$$

where T_{r2} and T'_{r2} are the operating time of R_2 in the network without and with DG unit integrated, respectively. As a result, the operating time of R_2 is shorter and false tripping of the relay in the network may occur.

Following the above analysis, the major impact on the overcurrent protection in a distribution network is that the operating time of the relays is changed due to the integration of the DG units into the adjacent network.

Reduced reach of impedance relays

The phenomena of the reduced reach of impedance relays due to the embedded power infeed is introduced in [97] [98]. The reach of an impedance relay is the maximum fault distance that triggers the relay in a certain impedance zone, or in a certain time. This maximum distance corresponds to a maximum fault impedance or a minimum fault current that is detected. Considering the situation shown in Figure 3.3, where a radial network R_3 linked to busbar b_1 is configured as an impedance relay. A DG unit (DG_2) is connected to b_2. If there in no DG integrated, the voltage measured by R_3 (U_{r3}) in case of a short circuit at F_3 is:

$$U_{r3} = I_{r3} Z_{1F_3} \tag{3.1.6}$$

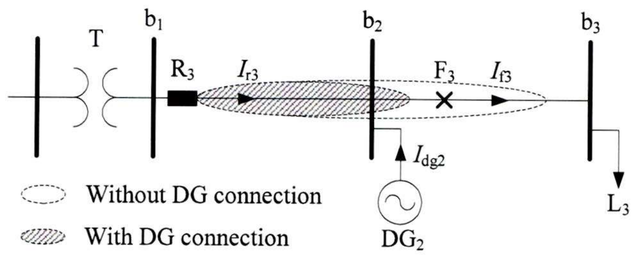


Figure 3.3: Reduced reach of an impedance relay.

where I_{r3} is the current detected by R_3, which equals to the fault current at F_3 (I_{f3}) and Z_{1F_3} is the line impedance from busbar b_1 to F_3. Therefore, the impedance measured by R_3 is:

$$Z_{r3} = \frac{U_{r3}}{I_{r3}} = Z_{1F_3} \tag{3.1.7}$$

However, when DG_2 is connected to b_2 and if the same fault occurs at F_3, the voltage measured by R_3 is increased due to an additional infeed at b_2, which is:

$$U'_{r3} = I_{r3}Z_{12} + I_{f3}Z_{2F_3} \tag{3.1.8}$$

where Z_{12} and Z_{2F_3} are the line impedances from b_1 to b_2 and b_2 to the fault location F_3, respectively. I_{f3} is the fault current at F_3 that is $I_{f3} = I_{r3} + I_{dg2}$. The impedance measured by R_3 changes as:

$$Z'_{r3} = \frac{U'_{r3}}{I_{r3}} = Z_{12} + Z_{2F_3} + \frac{I_{dg2}}{I_{r3}}Z_{2F_3} \tag{3.1.9}$$

Obviously, if DG_2 is connected to b_2, the impedance measured by R_3 is higher than the actual fault impedance due to the increased fault distance. Consequently, R_3 may trigger in higher grading time response corresponding to another distance zone. For certain relaying settings which were determined by manufacture, the fault has to be closer to the relay to operate it within the originally intended distance zone. The active area of the relay is therefore shortened, and therefore its reach is reduced. Consequently, the relay may not be tripped.

3.1.3 Related work

Currently many researchers focused on the improvement of the relaying schemes for protecting the DG integrated distribution networks using new technologies. For example, Giovanini [55] presented a study of wide area agents based on communication for primary and backup coordinated protection. Agents are used to give each protection component control capacity as well as the ability to communicate with other agents. In this protection system, agents were embedded in each of the conventional protection components to construct an IED relay. The agent searches for relevant information by communicating with other agents. Agent communication can take place at the same substation or at remote substations. This information can be used to detect primary and remote faults, relay misoperation, breaker failures, and to compensate such problems with a much better performance than can be done in traditional schemes. The preliminary simulation results show that the protection scheme maybe able to contribute toward the mitigation of wide-area disturbances and the power blackouts that frequently follow them.

Baxevanos [99] introduced an extended research in possible high-end protection and control methods for power distribution network protection and restoration. This work was intended to show the efficiency of combining modern IT techniques with the equipment provided by distribution automation evolving technology. Based on the proposed flexible and versatile multi-agent system structure, two specified cases were under investigation. The first one aimed at evaluating the abilities of the proposed system to work asynchronously and manage intranet and serial communication ports. The second case focused on the evaluation of the algorithms concerning a switching action of restoration. The simulation results indicate that the perspectives of the proposed system are quite promising.

In [100], a multi-agent approach to power system protection coordination was proposed. This system was developed based on the JADE platform, regarding the substation as one JADE Agent Container, which consists of a substation management agent, a number of relay agents, DG agents and equipment agents.

Coordination strategy was embedded in each relay agent to facilitate the other agents to be coordinated. The communication simulation was undertaken and the results show the feasibility of applying MAS in protection coordination. However, their work does not include a detailed analysis of application performance, or describe the agent implementation in detail. Moreover, only the simplest coordination strategy was considered and the simulation studies only focused on a simple fault situation in a distribution network.

3.2 Development of An Agent-based Relaying Scheme

This section presents the development of the agent-based relaying scheme for the protection of distribution networks with DG integrated. This scheme aims at improving the coordination between an integrated DG unit and its adjacent protection relays. As shown in Figure 3.4, a protection relay is composed of a sensing unit for measuring real-time current or voltage values, a processing unit for the determination of operating time according to the measurements and an operating unit for tripping or reclosing the protection relay. By applying the proposed scheme, the connection signals and capacity parameters of the DG unit can be transferred to its adjacent relays. Moreover, the scheme is able to regulate the real-time measurements collected from the sensing unit

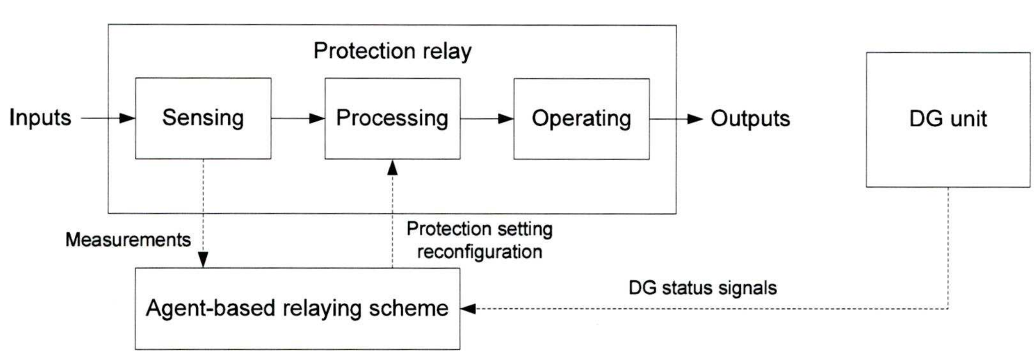


Figure 3.4: The agent-based relaying scheme.

of a relay, in accordance with the pre-defined control principles and the information received from the adjacent DG units. In this section, the structure of the proposed scheme is presented and the two specified agent modules, relay modules and DG modules, are described.

3.2.1 A structure of agent-based relaying scheme

As shown in Figure 3.5, the proposed agent-based relaying scheme contains two modules, a DG module installed at the terminal of a DG unit and a relay module embedded within a protection relay. The DG module collects the connection signals from the DG unit and transfer them to its adjacent relay modules. Moreover, the relay module is capable of regulating the real-time relay measurements and reconfiguring relay protection settings in accordance with the received information from the DG modules. Each module is composed of three software agents, a control agent ($\mathcal{A}_{\mathrm{CA}}$), a database agent ($\mathcal{A}_{\mathrm{DGA}}$) and a communication agent ($\mathcal{A}_{\mathrm{ComA}}$), which are developed in the multi-agent framework based on the generic structure presented in Chapter 2. One agent employed in different modules is specified to provide different functions. For example, in the relay module, the control agent is to collect and regulate the

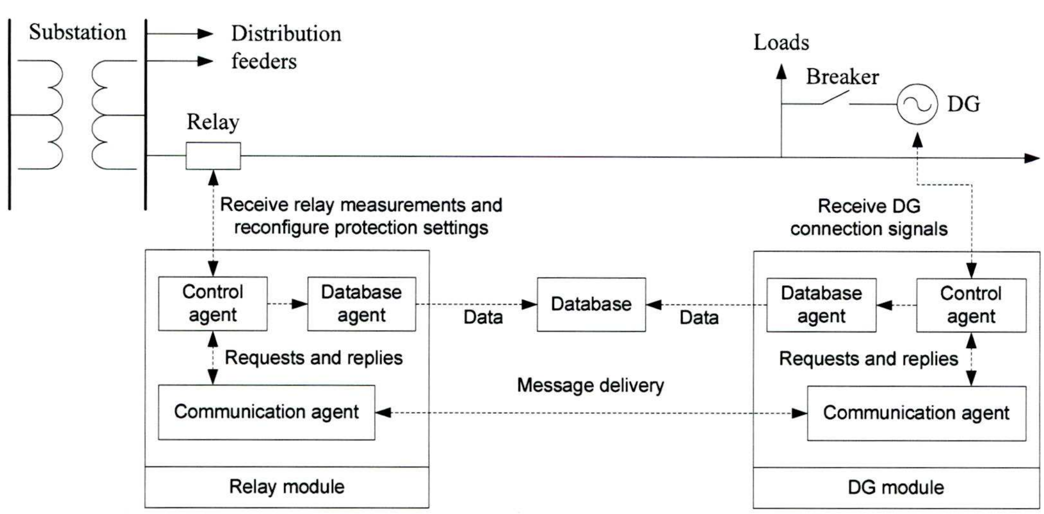


Figure 3.5: A structure of the agent-based relaying scheme.

real-time measurements, while in the DG module, it is capable of receiving the DG connection signals and its capacity parameters. Furthermore, communication between the two modules is based on the combined FIPA and UDP protocols introduced in Section 2.4.

3.2.2 DG module

The DG module, defined as $\mathcal{M}_{\text{DG}}$, can be expressed as:

$$\mathcal{M}_{\text{DG}} \supset \{\mathcal{A}_{\text{CA\_DG}}, \mathcal{A}_{\text{DBA\_DG}}, \mathcal{A}_{\text{ComA\_DG}}\}$$

where, $\mathcal{A}_{\text{CA\_DG}}$, $\mathcal{A}_{\text{DBA\_DG}}$ and $\mathcal{A}_{\text{ComA\_DG}}$ denote the control agent, database agent and communication agent employed in $\mathcal{M}_{\text{DG}}$, respectively. Three main functions are provided by this module, including collecting DG connection signals and capacity parameters, transferring these signals to the adjacent relay modules and managing DG operational data in a database.

Collect DG connection signals

Based on the generic structure of a control agent described in Section 2.3, $\mathcal{A}_{\text{CA\_DG}}$ is specified in $\mathcal{M}_{\text{DG}}$ to provide this function. The control principles of $\mathcal{A}_{\text{CA\_DG}}$ allows it to receive real-time connection signals and output capacities from the DG unit. Moreover, a parameter *DG_status* is defined to indicate the current DG connection status, which is presented as:

$$DG\_status = \begin{cases} \text{OPEN} & \text{if a DG unit is connected} \\ \text{CLOSE} & \text{if a DG unit is disconnected} \end{cases}$$

In Algorithm 1, the principle of transferring the DG connection signals and parameters is defined. Following this algorithm, once DG connection status is changed, the parameter *DG_status* is transferred by both $\mathcal{A}_{\text{DBA\_DG}}$ and $\mathcal{A}_{\text{ComA\_DG}}$ for saving DG operational data and informing the adjacent relay modules, respectively.

Algorithm 1 DG connection signal transfer

Input

t: sampling time

DG_status(t): connection status of DG unit at time t

DG_output(t): output current value of DG unit at time t

ID_ComA: identification number of $\mathcal{A}_{\text{ComA\_DG}}$

ID_DBA: identification number of $\mathcal{A}_{\text{DBA\_DG}}$

Begin

IF	DG_status(t + 1) $\neq$ DG_status(t)
THEN	Transfer(DG_status(t+1), DG_output(t+1), *ID_DBA*, *ID_ComA*)
ELSE IF	DG_status(t + 1) = DG_status(t)
THEN	Transfer(DG_status(t+1), DG_output(t+1), *ID_DBA*)

End

Send DG connection signals to relay modules

Based on the generic agent structure, $\mathcal{A}_{\text{ComA\_DG}}$ in the DG module maintains the knowledge for the communication with $\mathcal{A}_{\text{ComA\_R}}$ employed in the adjacent relay modules. In particular, $\mathcal{A}_{\text{ComA\_DG}}$ can be defined as:

$$\mathcal{A}_{\text{ComA\_DG}}(ID, DG\_status, DG\_output, Num\_\mathcal{A}_{\text{Up\_ComA\_R}}, ID\_\mathcal{A}_{\text{Up\_ComA\_R}}, Num\_\mathcal{A}_{\text{Dn\_ComA\_R}}, ID\_\mathcal{A}_{\text{Dn\_ComA\_R}})$$

where the identification number of $\mathcal{A}_{\text{ComA\_DG}}$ is defined as *ID*, which can be expressed as *ComA_DG1*, *ComA_DG2*, *etc.* *DG_status* and *DG_output* are parameters received from $\mathcal{A}_{\text{CA\_DG}}$. Moreover, $Num\_\mathcal{A}_{\text{Up\_ComA\_R}}$ and $ID\_\mathcal{A}_{\text{Up\_ComA\_R}}$ are the number and ID of $\mathcal{A}_{\text{ComA\_R}}$ in the adjacent upstream relay modules, while $Num\_\mathcal{A}_{\text{Dn\_ComA\_R}}$ and $ID\_\mathcal{A}_{\text{Dn\_ComA\_R}}$ are the number and ID of $\mathcal{A}_{\text{ComA\_R}}$ employed in the downstream relay module neighbours.

Following the maintained communication rules, if the parameters *DG_status* and *DG_output* are received from $\mathcal{A}_{\text{CA\_DG}}$, an ACL message is generated by $\mathcal{A}_{\text{ComA\_DG}}$ and sent to both $\mathcal{A}_{\text{Dn\_ComA\_R}}$ and $\mathcal{A}_{\text{Up\_ComA\_R}}$. The contents of this message include "*ID, DG_status, DG_output*", and the performative is INFORM. Furthermore, "AGREE" replies are received from $\mathcal{A}_{\text{Dn\_ComA\_R}}$ and

$\mathcal{A}_{\text{Up\_ComA\_R}}$ if they accept the messages. Otherwise, if the message can not be recognised or delivered, "REFUSE" or "FAILURE" messages are received.

Record DG unit operating data

$\mathcal{M}_{\text{DG}}$ also maintains a connection with a database for managing the operational data of the DG unit. Particularly, $\mathcal{A}_{\text{DBA\_DG}}$ is able to add or extract the real-time data collected by $\mathcal{A}_{\text{CA\_DG}}$ into and from the database, which can be utilised by engineers for monitoring the overall condition of the distribution system.

3.2.3 Relay module

The relay module, defined as $\mathcal{M}_{\text{R}}$, is composed of three generic agents, such as $\mathcal{A}_{\text{CA\_R}}$, $\mathcal{A}_{\text{DBA\_R}}$ and $\mathcal{A}_{\text{ComA\_R}}$. The expression of $\mathcal{M}_{\text{R}}$ is:

$$\mathcal{M}_{\text{R}} \supset \{\mathcal{A}_{\text{CA\_R}}, \mathcal{A}_{\text{DBA\_R}}, \mathcal{A}_{\text{ComA\_R}}\}$$

The main capabilities of the relay module are receiving DG connection signals and parameters from DG modules, collecting and regulating relay measurements in real-time and organising relay operational data in a database.

Receive DG connection signals

Based on the developed generic agent structure, $\mathcal{A}_{\text{ComA\_R}}$ employed in this module is capable of receiving ACL messages from $\mathcal{A}_{\text{ComA\_DG}}$ and taking response to each received message. The knowledge upheld by $\mathcal{A}_{\text{ComA\_R}}$ is:

$$\mathcal{A}_{\text{ComA\_R}}(ID, DG\_status, DG\_output, Num\_\mathcal{A}_{\text{ComA\_DG}}, ID\_\mathcal{A}_{\text{ComA\_DG}})$$

where, the identification number of $\mathcal{A}_{\text{ComA\_R}}$ is defined as *ID*, which can be expressed as *ComA_R1*, *ComA_R2*, *etc.* *DG_status* and *DG_output* are the received from $\mathcal{A}_{\text{ComA\_DG}}$. $Num\_\mathcal{A}_{\text{ComA\_DG}}$ and $ID\_\mathcal{A}_{\text{ComA\_DG}}$ are the number and ID of $\mathcal{A}_{\text{ComA\_DG}}$ employed in the adjacent DG modules.

In addition, the contents of the received message include "*ID_ComA_DG, DG_status, DG_output*" and the performative of this message is INFORM. This message is transferred to $\mathcal{A}_{\mathrm{CA\_R}}$ for the regulation of relay real-time measurements. Furthermore, an "AGREE" message is replied if the received message is accepted. Otherwise, if the message can not be recognised, a "REFUSE" message is returned back to $\mathcal{A}_{\mathrm{ComA\_DG}}$.

Collect and regulate relay measurements

One of the most important functions provided by the relay module is to collect and regulate the real-time relay measurements in accordance with the integration of a DG unit. This task is undertaken by $\mathcal{A}_{\mathrm{CA\_R}}$. Based on its generic structure introduced in Section 2.3, one of the control principles of $\mathcal{A}_{\mathrm{CA\_R}}$ is to collect relay measurements in runtime. A parameter *RT_value* indicating the real-time relay measurement values is defined. It is able to compare this value after each sampling with the previous one and calculate the value changes (*Value_change*). *RT_value* is also forwarded to $\mathcal{A}_{\mathrm{DBA\_R}}$ for the management of relay operational data.

In addition, as introduced in Section 3.1, a protection relay may be tripped incorrectly due to the integration of DG, since relay measurements may be affected. In this case, $\mathcal{A}_{\mathrm{CA\_R}}$ is capable of regulating the measurements following the control principle described in Algorithm 2.

Manage relay operational data

The relay module also provide a function for managing relay operational data in a database. $\mathcal{A}_{\mathrm{DBA\_R}}$ maintains the connection with the database for saving relay tripping or reclosure signals collected by $\mathcal{A}_{\mathrm{CA\_R}}$. Furthermore, it is able to extract the historical data from the database for monitoring relay operations in the overall distribution system.

3.3 Simulation System

Simulations studies are carried out to evaluate the performance of the proposed agent-based relaying scheme for the protection of a distribution network

Algorithm 2 Relay measurement regulation

Input

t:	sampling time
RT_value(t):	real-time measurement at time t
DG_status:	received DG connection signal
DG_output:	received DG output parameter
Reg_value	regulated measurement

Begin

Set $Value\_change = RT\_value(t) - RT\_value(t-1)$

If DG_status = CLOSE

  If $|Value\_change| < DG\_output$

  Then $Reg\_value = RT\_value(t) - Value\_change$

  Else $Reg\_value = 0$

Else if DG_status = OPEN

  If $|Value\_change| < DG\_output$

  Then $Reg\_value = RT\_value(t) + Value\_change$

  Else $Reg\_value = 0$

End

with DG integrated. In this section, a developed distribution network model and an agent communication network model are introduced, followed by descriptions of three specified simulation cases.

3.3.1 A distribution network model

A single line diagram of an IEEE 13-bus radial distribution network model, shown in Figure 3.6, is established in the PSCAD/EMTDC programme environment. A voltage source, as a power supplier, provides 100 MVA output power and 66 kV constant output voltage. The impedance of the voltage source is 5 Ω and the phase angle is 80°. The protection relays installed in the network are configured as time-overcurrent (TOC) relays that follow IEEE Extremely Inverse TOC Curves [96]. The operating time of breakers and reclosers employed in this model are configured as 20 ms. Moreover, the distances between

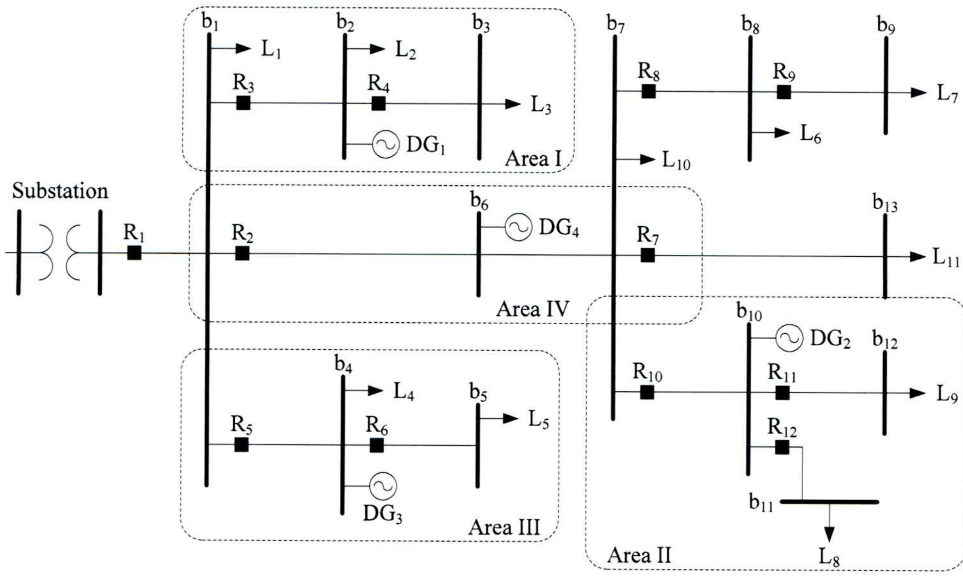


Figure 3.6: A distribution network model built in PSCAD/EMTDC.

the busbars in the distribution system are listed in Table 3.1.

Table 3.1: Distance between the busbars.

Busbar	Distance	Busbar	Distance	Busbar	Distance
b_1 to b_2	10 km	b_2 to b_3	5 km	b_1 to b_4	5 km
b_4 to b_5	15 km	b_1 to b_6	10 km	b_6 to b_7	10 km
b_7 to b_8	15 km	b_8 to b_9	15 km	b_7 to b_{10}	15 km
b_{10} to b_{11}	15 km	b_{10} to b_{12}	15 km		

Four DG units (DG_1, DG_2, DG_3 and DG_4) are connected to different busbars in the distribution network. Each DG unit is comprised of four individual wind power generators, providing 8 MVA output power and a three-phase constant current. Figure 3.7 displays a detailed construction of the wind power generator built in the PSCAD/EMTDC programme environment.

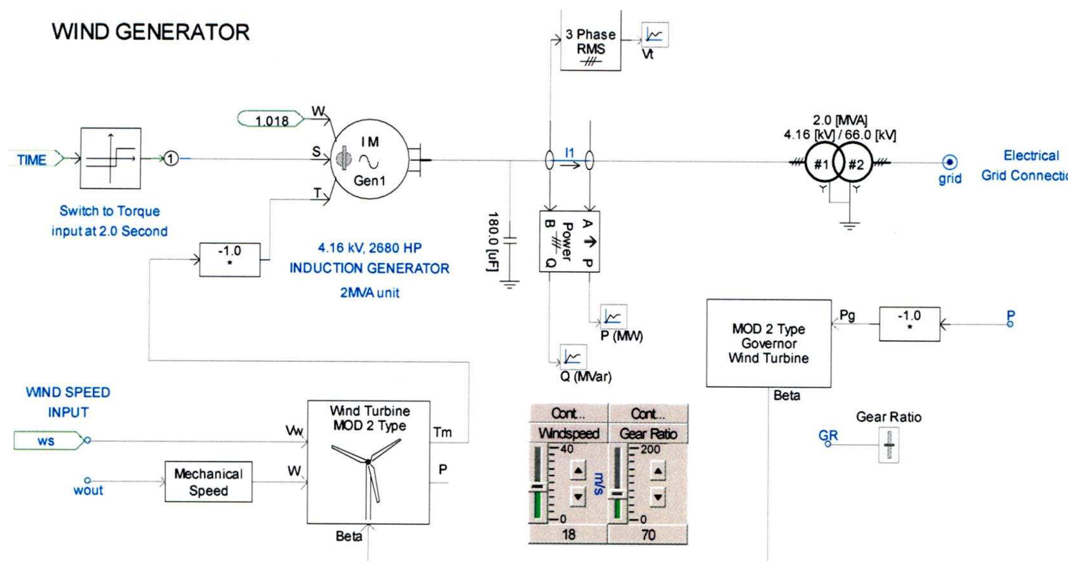


Figure 3.7: A simulation model of a wind generator.

3.3.2 An agent communication network model

An agent communication network is established based on the LAN in the e-Automation architecture for evaluating the performance of agent coordination in the proposed scheme. As shown in Figure 3.8, the relay and DG modules are installed in the PXI computers[1] connected to separate 10/100 Mbps Ethernet switches and a PC running the JADE agent platform is used as an agent server. Data utilised by the relay and DG modules are generated from each node of the distribution network and saved in a database that is executed in the agent server computer.

As the requirements of protecting a distribution network introduced in Section 2.4.1, messages should be transmitted among agents within a very short time frame. A possible solution concerning the real-time communication difficulties of agent coordination is proposed in the developed multi-agent framework presented in Chapter 2. Based on this framework, a FIPA-compliant agent platform and a UDP-based communication platform are combined to-

[1]PCI eXtensions for Instrumentation (PXI) is a modular instrumentation platform introduced by National Instruments.

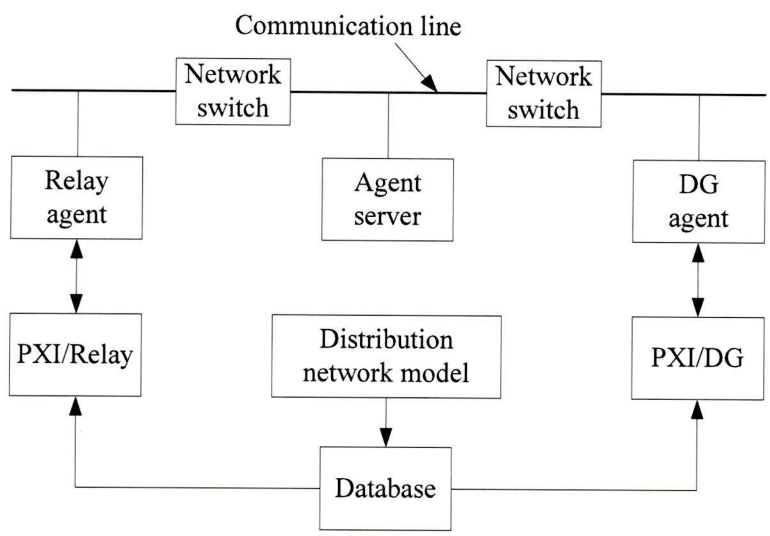


Figure 3.8: The agent communication network.

gether. Thus, in the agent communication network built for the simulation study, the JADE platform is extended to combine with a gateway agent, which allows any number of independent agents supporting the UDP protocol to join the system. Each agent employed in the proposed scheme is able to listen to messages on a UDP port and a broadcast port, which allows both peer-to-peer and broadcast messaging.

3.3.3 Simulation cases

Three cases are designed based on the distribution network model for evaluating the performance of the proposed scheme in handling different protection issues. In particular, three scenarios are carried out in each case for the comparison of the operating times of the relays in different situations, which are listed as follows:

- **Scenario 1:** No DG unit is integrated with the network when a fault occurs.
- **Scenario 2:** A DG unit is connected and then the same fault appears at the same time.

- **Scenario 3:** The proposed agent-based relaying scheme is applied to coordinate the operations of the integrated DG unit and its adjacent relays, then the same fault occurs at the same time.

Case I

This case is to investigate the performance of the proposed scheme in handling a protection issue when a fault occurs in the upstream network of a integrated DG unit. Figure 3.9 illustrates the area I of the established distribution network model. In this area, two TOC relays (R_3 and R_4) are connected to the busbars b_1 and b_2, respectively and a DG unit (DG_1) is to be connected to b_2. In this case, a single-phase-to-ground fault is set to occur at the point F_1 which is 1 km away from b_1 and the fault resistance is 20 Ω.

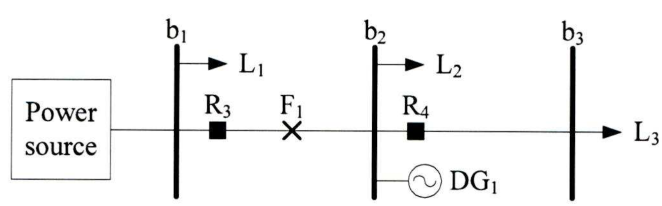


Figure 3.9: Distribution network model in simulation case I of agent-based relaying scheme.

Case II

The second case is carried out in area II of the distribution network, displayed in Figure 3.10, for the evaluation of the proposed scheme in handling a multiple faults issue. Three TOC relays (R_{10}, R_{11} and R_{12}) are installed in this area and a DG unit (DG_2) is to be connected to b_{10}. The first fault occurs at the point F_2, which is 5 km away from b_{10}. It is a three-phase-to-ground fault with 10 Ω fault resistance. The second one is a phase-to-phase fault that appears at the point F_3 that is 2 km away from b_{10} and the fault resistance is 5 Ω.

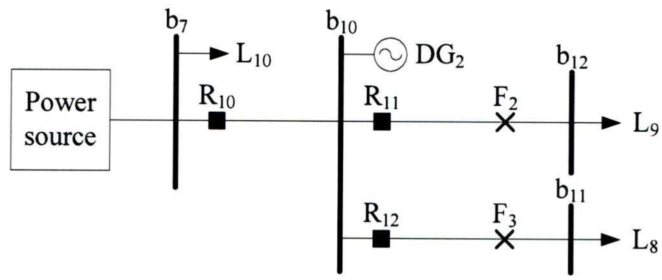


Figure 3.10: Distribution network model in simulation case II of agent-based relaying scheme.

Case III

This case is to investigate the protection issue of multiple DG units connected in both the upstream and downstream network. As shown in Figure 3.11, two TOC relays (R_5 and R_6) are connected to b_1 and b_4, respectively and two DG units are to be connected to b_4 and b_5. A single-phase-to-ground fault occurs at the point F_4, which is 5 km away from b_4 and the fault resistance is 15 Ω.

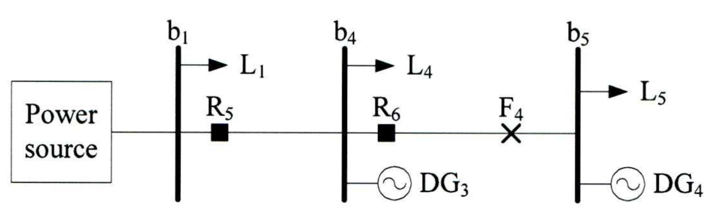


Figure 3.11: Distribution network model in simulation case III of agent-based relaying scheme.

3.4 Simulation Results

3.4.1 Single fault scenario

Agent coordination

At the initialisation stage, two relay modules ($\mathcal{M}_{\mathrm{R3}}$ and $\mathcal{M}_{\mathrm{R4}}$) and one DG module ($\mathcal{M}_{\mathrm{DG1}}$) are executed. Following case I described in the previous section, when DG_1 is connected to b_2, $\mathcal{M}_{\mathrm{DG1}}$ is executed and $\mathcal{A}_{\mathrm{ComA\_DG1}}$ sends two messages to $\mathcal{A}_{\mathrm{ComA\_R3}}$ and $\mathcal{A}_{\mathrm{ComA\_R4}}$, respectively. The contents of the message include "*ComA_DG1, CLOSE, 0.06*". As shown in Figure 3.12, when the message is received, both $\mathcal{A}_{\mathrm{ComA\_R3}}$ and $\mathcal{A}_{\mathrm{ComA\_R4}}$ reply "AGREE" messages to $\mathcal{A}_{\mathrm{ComA\_DG1}}$ if the correct operations are carried out by both of them. The time consumption of a message delivered from $\mathcal{A}_{\mathrm{ComA\_DG1}}$ to $\mathcal{A}_{\mathrm{ComA\_R3}}$ in this simulation is approximately 34 ms.

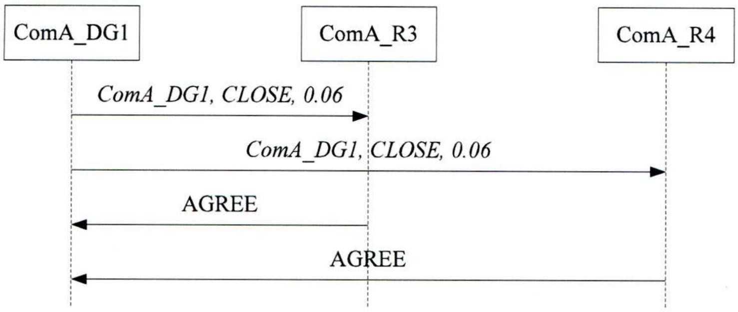
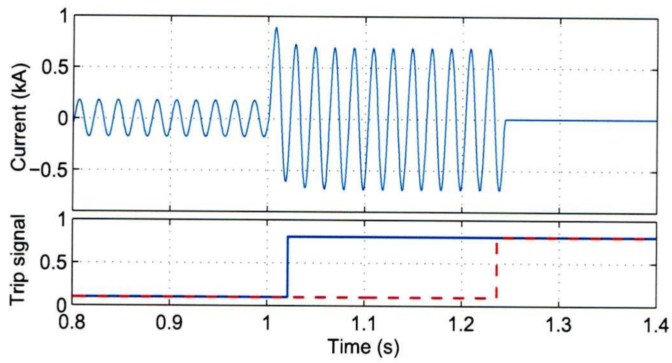


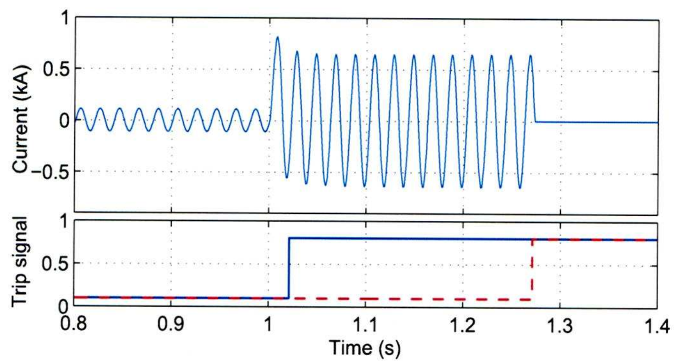
Figure 3.12: Agent coordination in case I of agent-based relaying scheme.

Relay operating time

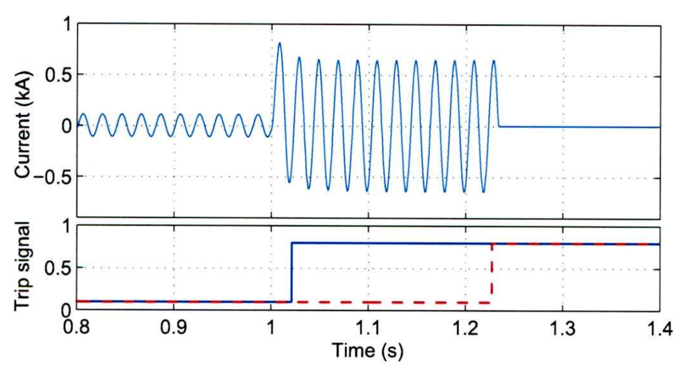
In this case, a phase A-to-ground fault is set to occur at the point F_1 at 1 s. Figure 3.13(a) displays the waveform of the phase-A current measured by R_3. If no DG unit is integrated with the network, the RMS value measured at the steady state is approximately 0.126 kA and the fault current value is 0.489 kA. Following its characteristics, R_3 is tripped at 1.235 s.



(a) R_3 measurement and trip signal in scenario 1 of agent-based relaying scheme.



(b) R_3 measurement and trip signal in scenario 2 of agent-based relaying scheme.



(c) R_3 measurement and trip signal in scenario 3 of agent-based relaying scheme.

Figure 3.13: Simulation results in case I of agent-based relaying scheme.

When DG_1 is connected to b_2, the RMS value of phase-A current measured by R_3 at the steady state reduces to 0.078 kA. If the same fault occurs at 1 s, the measured fault current value is also changed to 0.448 kA, which is

0.041 kA lower than the real fault current value. Consequently, R_3 is tripped at 1.270 s and the operating time of R_3 is prolonged by approximately 35 ms. Figure 3.13(b) illustrates the current waveform measured by R_3 and its operational signals.

In addition, if the proposed agent-based relaying scheme is applied, a message is sent from the DG module to both relay modules when DG_1 is connected. In this case, $\mathcal{A}_{CA\_R3}$ records the decrements of phase-A current measured by R_3 and sets the parameter $Value\_change$ as -0.048. Following Algorithm 2, $Value\_change$ is subtracted from the real measurements. When the same fault appears again at 1 s, the fault current value processed by R_3 is changed to 0.496 kA. Accordingly, R_3 is tripped at 1.226 s. The delayed tripping of R_3 in the second case can be avoided. In Table 3.2, the detailed operating times of R_3 in the different simulation situations are listed.

Table 3.2: Operating time of R_3 in case I of agent-based relaying scheme.

Scenario	fault occurrence	Tripping time	Operating time
1	1 s	1.235 s	215 ms
2	1 s	1.270 s	250 ms
3	1 s	1.226 s	206 ms

3.4.2 Multiple faults scenario

Agent coordination

In the second case, three relay modules ($\mathcal{M}_{R10}$, $\mathcal{M}_{R11}$ and $\mathcal{M}_{R12}$) and one DG module ($\mathcal{M}_{DG2}$) are initialised. As shown in Figure 3.14, when DG_2 is integrated, three same messages are sent to $\mathcal{A}_{ComA\_R10}$, $\mathcal{A}_{ComA\_R11}$ and $\mathcal{A}_{ComA\_R12}$, respectively. The contents of a message contain "*ComA_DG2, CLOSE, 0.04*". When the received message is accepted and the correct operation is carried out, an "AGREE" message is replied by each communication agent employed in the relay module. Moreover, it takes approximately 34 ms for transferring a message from $\mathcal{A}_{\text{ComA\_DG2}}$ to $\mathcal{A}_{\text{ComA\_R10}}$, $\mathcal{A}_{\text{ComA\_R11}}$ and $\mathcal{A}_{\text{ComA\_R12}}$, respectively.

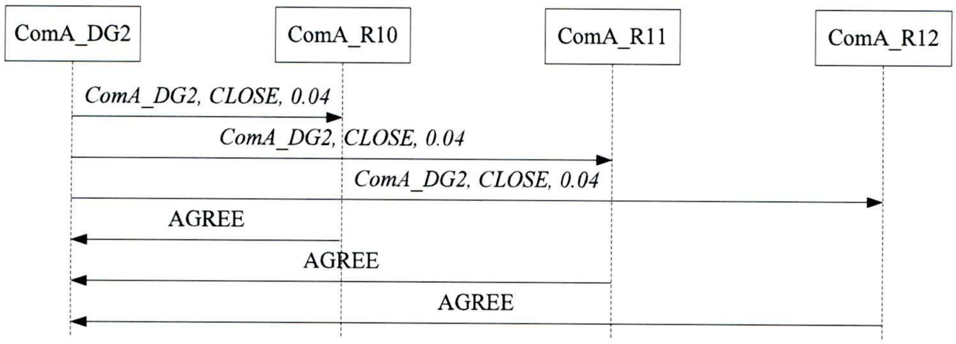
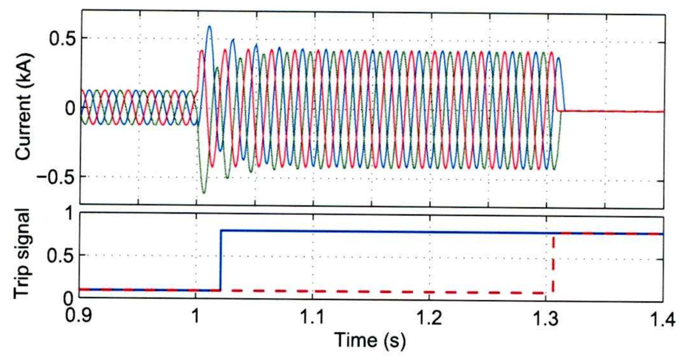


Figure 3.14: Agent coordination in case II.

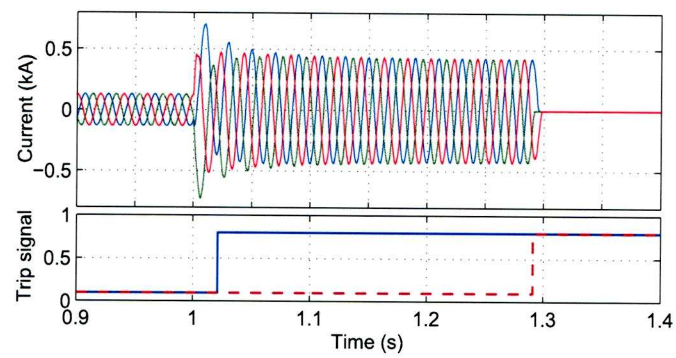
Relay operating time

In this simulation, two faults, a three phase-to-ground fault and a phase-to-phase fault, occur at the point F_2 and F_3, respectively. In particular, the first fault is set to appear at 1 s, while the second one occurs at 1.35 s. As displayed in Figures 3.15(a) and 3.16(a), at the steady state, the RMS values of the currents measured by R_{11} and R_{12} in the network without DG unit embedded are both 0.089 kA. When the first fault occurs, the RMS value of the fault current is 0.300 kA. R_{11} is then tripped at 1.305 s. When the second fault appears, the fault current value is 0.318 kA and R_{12} is tripped at 1.620 s following its characteristics.

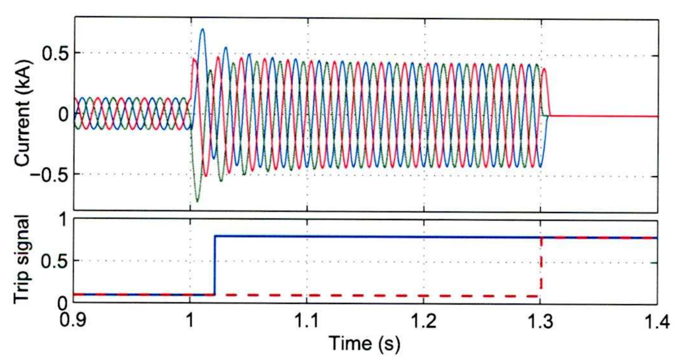
If DG_2 is embedded within the network, the steady state current value measured by both R_{11} and R_{12} are increased to 0.096 kA. If the same three phase-to-ground fault occurs at 1 s, the fault current value measured by R_{11} is changed to 0.310 kA, which is 0.010 kA higher than the real fault current value. In this case, R_{11} is tripped at 1.290 s, which 15 ms earlier than the normal tripping time. Moreover, if the same three-phase fault appears at 1.35 s, the fault current value measured by R_{12} is also increased as 0.327 kA. Following its characteristics, the operating time of R_{12} is changed to 250 ms, which is shorter



(a) R_{11} measurement and trip signal in scenario 1 of agent-based relaying scheme.



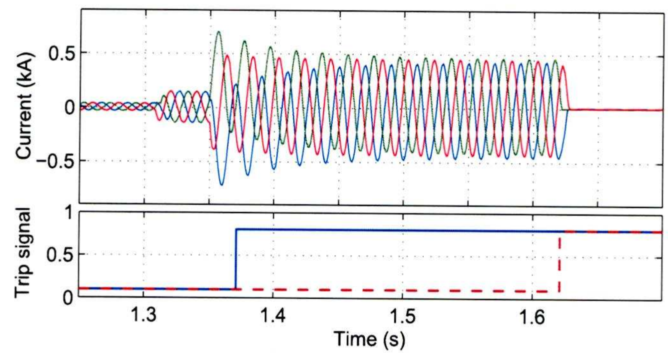
(b) R_{11} measurement and trip signal in scenario 2 of agent-based relaying scheme.



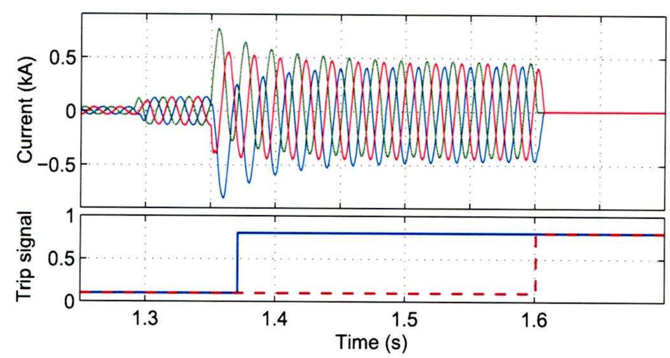
(c) R_{11} measurement and trip signal in scenario 3 of agent-based relaying scheme.

Figure 3.15: Current measurements of R_{11} in case II of agent-based relaying scheme.

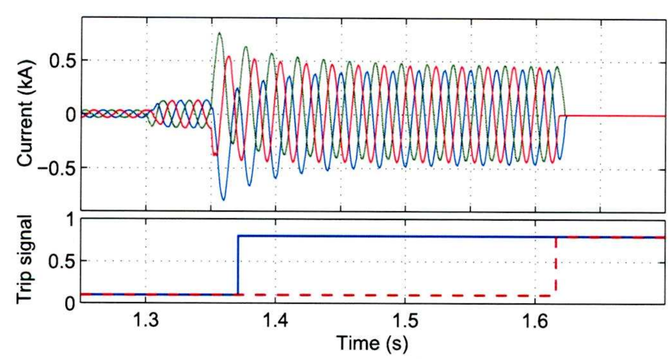
than the correct operating time. Obviously, both R_{11} and R_{12} are tripped at the wrong time when DG_2 is connected to their adjacent network. The three-phase current waveforms measured by R_{11} and R_{12} in this scenario are shown



(a) R_{12} measurement and trip signal in scenario 1 of agent-based relaying scheme.



(b) R_{12} measurement and trip signal in scenario 2 of agent-based relaying scheme.



(c) R_{12} measurement and trip signal in scenario 3 of agent-based relaying scheme.

Figure 3.16: Current measurements of R_{12} in case II of agent-based relaying scheme.

in Figures 3.15(b) and 3.16(b), respectively.

In the third scenario, the proposed scheme is applied to coordinate the operations of the relays. When DG_2 is connected, $\mathcal{M}_{DG2}$ is executed and all of the adjacent relay modules are informed. According to Algorithm 2, the measurement changes recorded by both R_{11} and R_{12}, are subtracted from the real-time measurements. In this case, when the two faults appear again, the fault current values used for processing are regulated as 0.303 kA and 0.319 kA, respectively. As illustrated in Figures 3.15(c) and 3.16(c), R_{11} is tripped at 1.302 s and the operating time of R_{12} is recovered as 266 ms, accordingly. Tables 3.3 and 3.4 demonstrate the detailed operating times of R_{11} and R_{12} respectively in the different scenarios.

Table 3.3: Tripping times of R_{11} in case II of agent-based relaying scheme.

Scenario	Fault occurrence	Tripping time	Operating time
1	1 s	1.305 s	305 ms
2	1 s	1.290 s	290 ms
3	1 s	1.303 s	303 ms

Table 3.4: Tripping times of R_{12} in case II of agent-based relaying scheme.

Scenario	Fault occurrence	Tripping time	Operating time
1	1.35 s	1.620 s	270 ms
2	1.35 s	1.600 s	250 ms
3	1.35 s	1.616 s	266 ms

3.4.3 Integration of multiple DG units

Agent coordination

In case II, more than one DG units are integrated with the network. Two relay modules ($\mathcal{M}_{\mathrm{R5}}$, $\mathcal{M}_{\mathrm{R6}}$) and two DG modules ($\mathcal{M}_{\mathrm{DG3}}$, $\mathcal{M}_{\mathrm{DG4}}$) are initialised at the beginning. Figure 3.17 displays the message transfers among the agents. A message, including "*ComA_DG3, CLOSE, 0.04*", is sent over to both $\mathcal{A}_{\mathrm{ComA\_R5}}$ and $\mathcal{A}_{\mathrm{ComA\_R6}}$ by $\mathcal{A}_{\mathrm{ComA\_DG3}}$, while another message is sent to both of the relay modules at the same time by $\mathcal{A}_{\mathrm{ComA\_DG4}}$ when it is initialised. Furthermore, "AGREE" messages are replied to $\mathcal{A}_{\mathrm{ComA\_DG3}}$ and $\mathcal{A}_{\mathrm{ComA\_DG4}}$ if the received messages are accepted and the correct operations are undertaken by $\mathcal{A}_{\mathrm{ComA\_R5}}$ and $\mathcal{A}_{\mathrm{ComA\_R6}}$. In particular, the average time consumption for message delivery between two agents is about 34 ms.

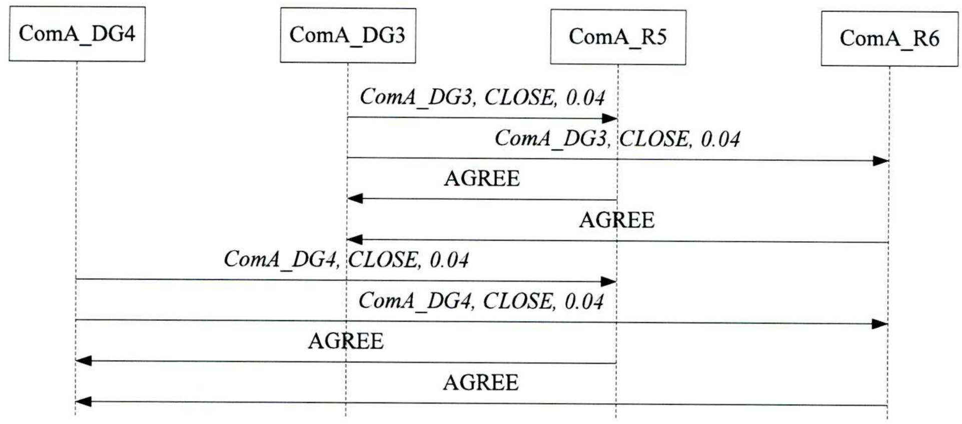


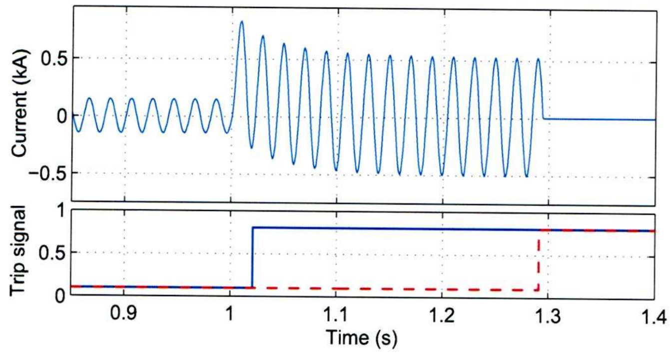
Figure 3.17: Agent coordination in case III of agent-based relaying scheme.

Relay operating time

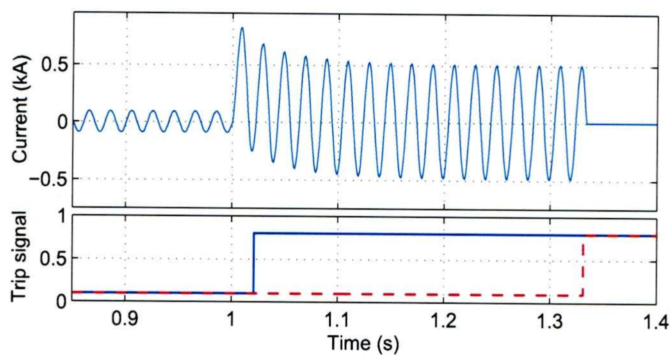
In this case, a phase-A-to-ground fault is set to occur at 1 s at the point F_4 in the network. Figure 3.18(a) displays the waveform of phase-A current measured by R_6 in the network without DG unit integrated. The RMS values of the phase-A current at steady state is 0.104 kA, while the fault current value is 0.368 kA. Following its characteristics, R_6 is tripped at 1.290 s.

When DG_3 and DG_4 are connected to the network, the current value measured by R_6 at the steady state is 0.075 kA. This has decreased by 0.029 kA. If the same fault occurs again, the fault current value is altered as 0.343 kA, which is 0.025 kA lower than the real fault current value. Consequently, R_6 is tripped at 1.330 s and the operating time of R_6 is prolonged to 330 ms. Figure 3.18(b) illustrates the current waveform measured by R_6 and its trip signal in this situation.

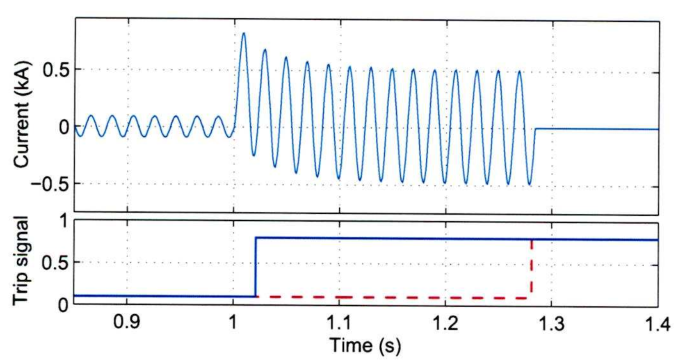
In order to avoid the delayed tipping of R_6, the proposed agent-based re-



(a) R_6 measurement and trip signal in scenario 1 of agent-based relaying scheme.



(b) R_6 measurement and trip signal in scenario 2 of agent-based relaying scheme.



(c) R_6 measurement and trip signal in scenario 3 of agent-based relaying scheme.

Figure 3.18: Simulation results in case III of agent-based relaying scheme.

laying scheme is applied. As introduced previously, when DG_3 is connected, a message is sent from $\mathcal{A}_{\text{ComA\_DG3}}$ to $\mathcal{A}_{\text{ComA\_R6}}$. At the same time, $\mathcal{A}_{\text{ComA\_R6}}$ receives another message from $\mathcal{A}_{\text{ComA\_DG4}}$ to indicate the integration of DG_4.

According to these two messages, $\mathcal{A}_{\mathrm{CA\_R6}}$ sets the parameter $Value\_change$ as -0.029 and subtracts it from the real measurement value. Following Algorithm 2, the measured current value at the steady state is regulated as 0.104 kA. If the same fault occurs again, the fault current value is adjusted as 0.372 kA. In this case, R_6 is tripped at 1.284 s and the operating time of R_6 is recovered as 284 ms. In Table 3.5, the operating times of R_6 in this case are demonstrated.

Table 3.5: Tripping times of R_6 in case III of agent-based relaying scheme.

Scenario	Fault occurrence	Tripping time	Operating time
1	1 s	1.290 s	290 ms
2	1 s	1.330 s	330 ms
3	1 s	1.284 s	284 ms

3.5 Evaluation and Discussion

3.5.1 Disconnection of a DG unit

In the simulation study, all of the cases are designed based on integrating the DG units with the distribution network. The proposed agent-based relaying scheme is also able to support for the coordination of relay operations when a DG unit is disconnected, because an agent can be emerged into or deregistered from the existing multi-agent system in runtime without any reconfiguration.

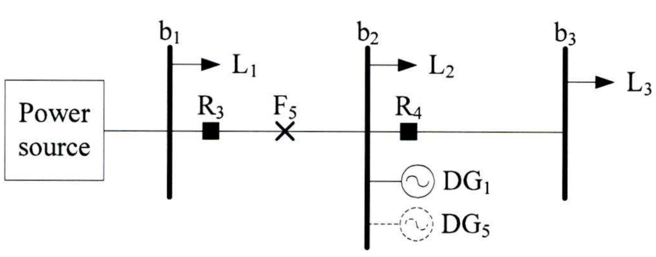
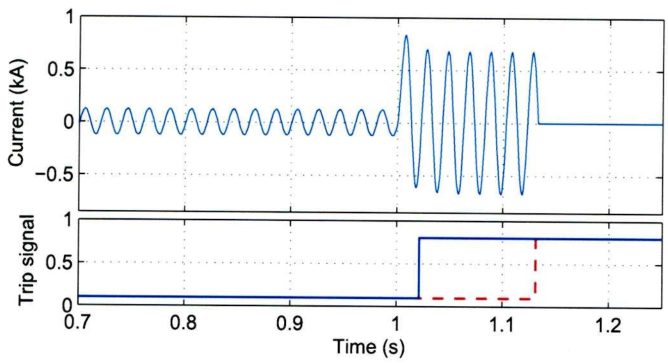
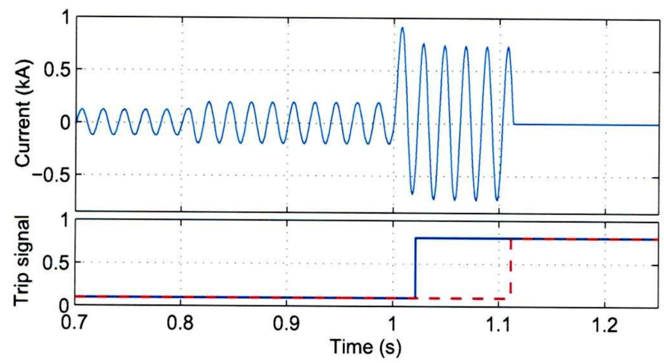


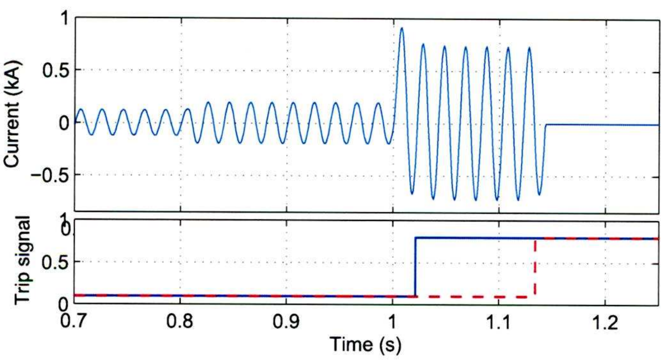
Figure 3.19: Area I of the distribution network.



(a) R_3 measurement and trip signal if DG_5 is connected.



(b) R_3 measurement and trip signal if DG_5 is disconnected.



(c) R_3 measurement and trip signal if DG_5 is disconnected and the proposed scheme is applied.

Figure 3.20: Simulation results of disconnection of DG_5.

In this case, another simulation case is carried out for investigating the performance of proposed scheme in handling a protection issue when a DG unit is disconnected with the distribution network.

Particularly, this simulation study is performed in area I of the distribution network model, shown in Figure 3.19, where two DG units (DG_1 and DG_5) are both connected to b_2. In this simulation, DG_5 is to be disconnected at 0.8 s and a phase-A-to-ground fault occurs at 1 s at the point F_5 which is 2 km away from b_1 and the fault resistance is 25 Ω.

As illustrated in Figure 3.20(a), if DG_5 is connected to b_2, the RMS value of phase-A current measured by R_3 at steady state is 0.087 kA, while the fault current is 0.478 kA. Following the characteristics of R_3, it is tripped at 1.130 s. The operating time of R_3 is about 130 ms. However, if DG_5 is disconnected at 0.8 s and the same fault occurs at 1 s, the measured current value between 0.8 s and 1 s is changed to 0.139 kA. This is a increase of 0.052 kA. Accordingly, the fault current value measured by R_3 is altered as 0.527 kA. Consequently, R_3 is tripped earlier at 1.110 s, which is displayed in Figure 3.20(b).

When the proposed agent-based relaying scheme is applied and DG_5 is disconnected at 0.8 s, a message is sent from $\mathcal{M}_{DG5}$ to both the upstream and downstream relay modules ($\mathcal{M}_{R3}$ and $\mathcal{M}_{R4}$). Specifically, the contents of this message contain "*ID_ComA_DG5, OPEN, 0.06*" and the performative is INFORM. After receiving this message, the measurement changes of R_3 is recorded by $\mathcal{A}_{DBA\_R3}$ and the parameter $Value\_change$ is set as 0.052. Following Algorithm 2, $Value\_change$ is added into the original measurements. The coordination processes between the agents are displayed in Figure 3.21.

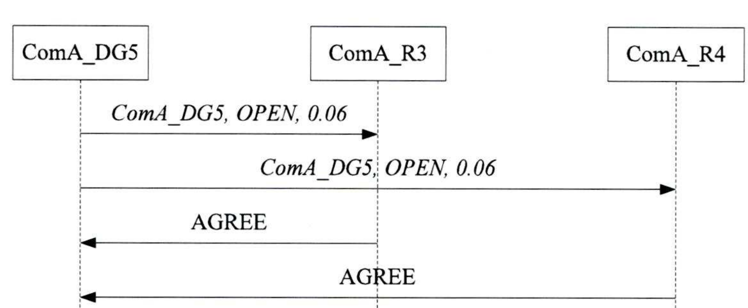


Figure 3.21: Agent coordination when DG_5 is disconnected.

If the same fault appears at 1 s, the measured fault current value is regulated as 0.475 kA, and accordingly R_3 is tripped at 1.133 s. The operating time of R_3 is recovered as 133 ms. The detailed simulation results of the operating times of R_3 in different situations are demonstrated in Table 3.6. As indicated by the results, the proposed agent-based relaying scheme is capable of coordinating the relay operations if an adjacent DG unit is disconnected from a distribution network in runtime.

Table 3.6: Operating time of R_3 if DG_5 is disconnected.

Scenario	Fault occurrence	Tripping time	Operating time
1	1 s	1.130 s	130 ms
2	1 s	1.110 s	110 ms
3	1 s	1.133 s	133 ms

3.5.2 Impedance protection

In the previous section, the performance of the proposed agent-based relay scheme in handling overcurrent protection issues is investigated. However, as introduced in Section 3.1.2, the integration of DG may not only affect overcurrent protection issues, but also may reduce the reach of impedance relays. In this case, another simulation study is carried out for the evaluation of the proposed scheme in impedance protection. Figure 3.22 displays the area IV of the distribution network model in which R_2 connected to b_1 is configured as a three-zone impedance relay.

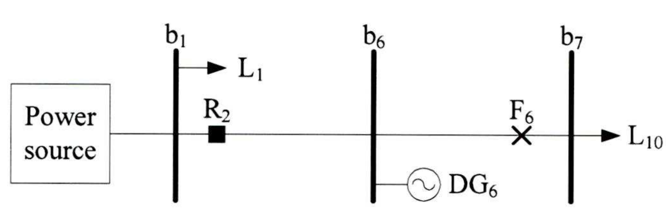


Figure 3.22: Area IV of the distribution network.

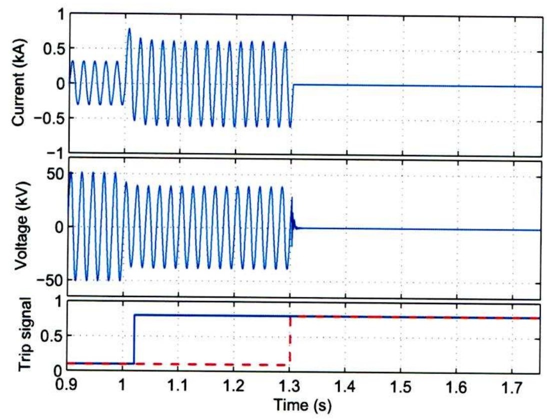
In this area, the distances between b_1 and b_6 and b_6 and b_7 are 10 km and 8 km, respectively. Specifically, zone 1 of R_2 reaches 80% of the distribution line between b_1 and b_6 (8 km), zone 2 of R_2 protects 150% of the distribution line between b_1 and b_6 (15 km) and zone 3 covers 120% of the distribution line between b_1 and b_7. The operating times of the three zones are set as 100 ms, 300 ms and 600 ms, respectively. In particularly, for a line-to-ground fault, the calculation of impedance is based on the following equation:

$$Z = \frac{V_{\text{phase}}}{I_{\text{phase}} + kI_0} \tag{3.5.1}$$

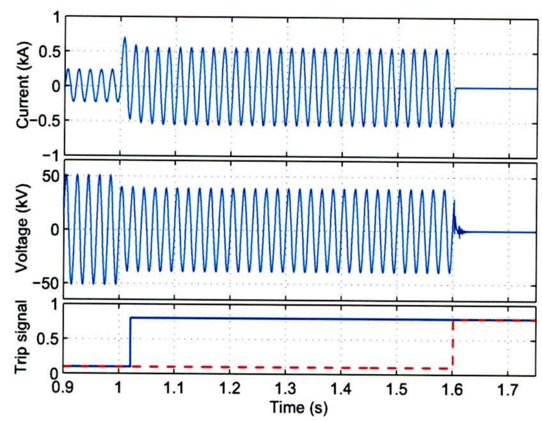
where V_{phase} and I_{phase} are the values of phase voltage and current and I_0 is the zero sequence current. Moreover, k is a constant for ground impedance, $k = \frac{Z_0 - Z_1}{Z_1}$, where Z_0 and Z_1 are the zero-sequence and positive-sequence impedance as seen from the location of the relay to the end of the protected zone, respectively. In this simulation network, $k = 2.75$.

In addition, a phase-A-to-ground fault with 30 Ω fault resistance is set to occur at 1 s at the point F_6 located 5 km away from b_6, which is just covered by the zone 2 of R_2. Three scenarios are considered in this simulation, which are the same as the scenarios described in Section 3.3.3. As shown in Figure 3.23(a), in the first simulation, the magnitudes of phase-A current and voltage measured by R_2 at the steady state are 0.220 kA and 36.14 kV. The fault current and voltage magnitudes are 0.437 kA and 27.47 kV, respectively. Accordingly, the fault impedance measured by R_2 in this situation is $Z = 16 + 42.4j$ and R_2 is tripped at 1.30 s following its characterises.

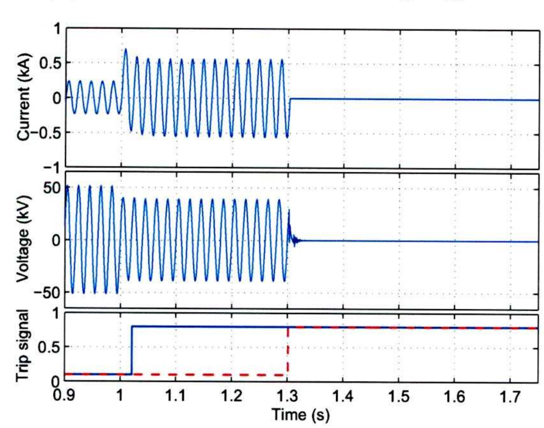
If a DG unit (DG_6) is connected to b_6, the current and voltage magnitude measurements are changed to 0.178 kA and 36.38 kV. If the same fault appears again, the measured fault current and voltage values are 0.399 kA and 27.72 kV, and therefore the fault impedance measured by R_2 is changed to $Z' = 13.84+44.71j$. Obviously, $|Z'| > |Z|$, and therefore the measured fault distance is out of the zone 2 of R_2. Consequently, R_2 is tripped at 1.6 s. Figure 3.23(b) illustrates the waveforms of the current and voltage measured by R_2 and its trip signal.



(a) R_2 measurement and trip signal.



(b) R_2 measurement and trip signal.



(c) R_2 measurement and trip signal.

Figure 3.23: Simulation results of impedance protection.

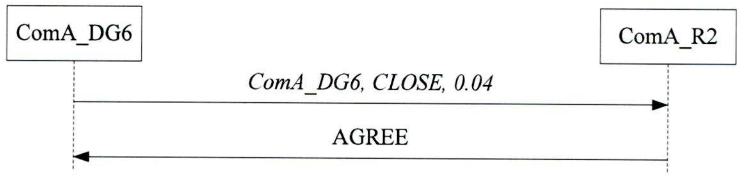


Figure 3.24: Agent coordination in impedance protection.

If the proposed agent-based relaying scheme is applied and DG_6 is connected to b_6, a DG module ($\mathcal{M}_{DG6}$) and a relay module $\mathcal{M}_{R2}$ are executed. As shown in Figure 3.24, a message is sent from $\mathcal{M}_{DG6}$ to inform $\mathcal{M}_{R2}$ of the integration of DG_6. The contents of this message include "*ID_ComA_DG6, CLOSE, 0.04*" and the performative is INFORM. According to the received message, $\mathcal{A}_{CA\_R2}$ records the measurement changes of both current and voltage and sets the parameters $Value\_change\_current$ and $Value\_change\_voltage$ as -0.042 and 0.24, respectively. Moreover, following Algorithm 2, both $Value\_change\_current$ and $Value\_change\_voltage$ are subtracted from the real measurement values in run time. If the same fault occurs again, the fault current and voltage values are regulated as 0.439 kA and 27.71 kV. In this case, the fault impedance calculated by R_2 is $Z'_{\text{Agent}} = 16.12 + 42.28j$, and obviously, $|Z'_{\text{Agent}}| < |Z|$. As a result, the fault distance measured by R_2 is in the zone 2 again, and the tripping time of R_2 is recovered as 300 ms, which is displayed in Figure 3.23(c). In Table 3.7 the operating times of R_2 in the three simulations are listed in detail.

Table 3.7: Operating time of R_2.

Scenario	Fault occurrence	Tripping time	Operating time
1	1 s	1.300 s	300 ms
2	1 s	1.600 s	600 ms
3	1 s	1.300 s	300 ms

3.5.3 Timing performance of agent communication in proposed scheme

As introduced in Section 2.4, time consumption of agent communications is one of the critical issues for power system automation, especially for protecting a power system. In this case, the timing performance of the proposed agent-based relaying scheme is under investigation. Particularly, two experiments are carried out based on the established agent communication model, shown in Figure 3.25, for the evaluation of the time consumption of message delivery among the agents employed in the proposed scheme. Three computers are utilised including two client computers and an agent server computer, which are configured as Pentium IV 2.8 GHz. Moreover, all of the computers are connected to the same 100BaseT LAN through a network switch. The protocol used for these interactions was based on UDP/IP. In addition, the timing performance in the experiments is defined as the time cost of a message being sent from a DG module and received by a relay module.

In the first experiment, only one DG module ($\mathcal{M}_{DG1}$) is initialised and the

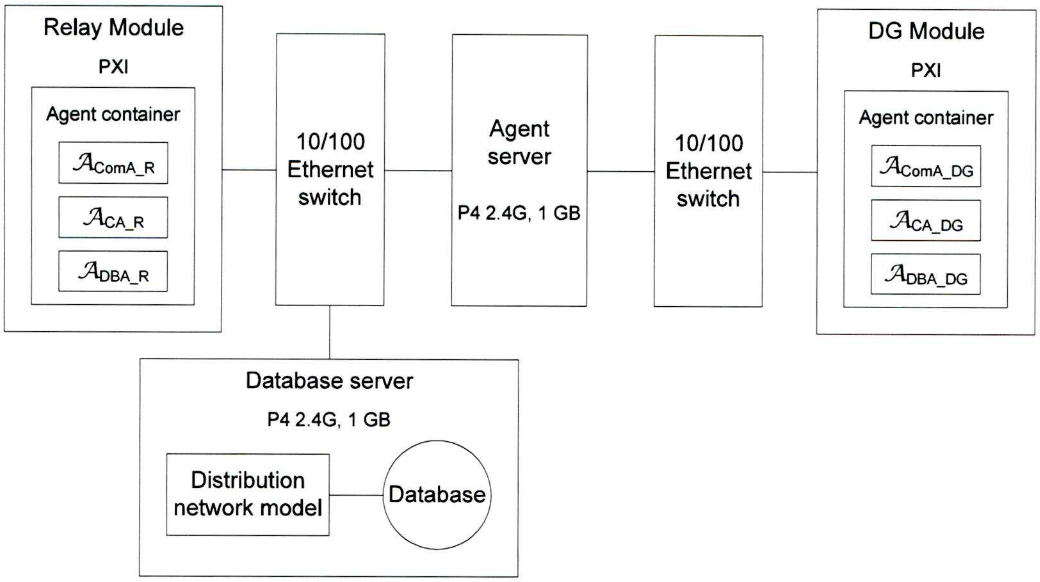


Figure 3.25: Agent communication network.

number of executed relay modules are varied from 1 to 10 in increments of one each time. Furthermore, the number of messages sent by the DG module at the same time also increases from 1 to 10. The contents of each message include "*ID_ComA_DG1, CLOSE, 0.06*" and the performative is INFORM. Table 3.8 demonstrates the time consumption of the messages transferred from one DG module to the relay modules.

Table 3.8: Time consumption of agent communication in experiment 1.

	The number of sent messages	The number of relay modules				
		2	4	6	8	10
Time (ms)	1	31	63	63	94	107
	2	62	78	109	131	140
	4	78	110	125	156	187
	6	91	128	153	195	221
	8	102	140	188	219	282
	10	110	157	219	282	344

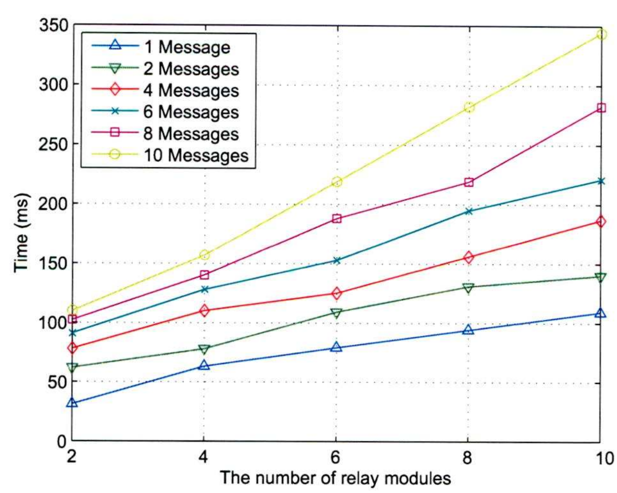


Figure 3.26: Timing performance of agent communication in experiment 1.

As seen from the table, the timing performance of the proposed scheme is related to the number of relay modules and the amount of messages sent by the

DG module at the same time. Specifically, if the number of the executed relay modules rises, the total time costs of message delivery are increased, because there are more messages being sent over by the DG modules. Furthermore, the time consumption raises along with the increments of the number of messages sent by the DG unit at the same time. The comparison of the timing costs in this experiment is displayed in Figure 3.26.

In addition to this, another experiment is carried out to investigate the timing performance of the scheme in the situation that more than one DG modules send messages to the relay modules simultaneously. In this experiment, up to 10 DG modules are employed and only one message is sent by one DG module. The number of the relay modules for receiving the messages is changed from 5 to 30 in increment of 5 each time. In particular, the contents of a message contain "*ID_ComA_DGi, CLOSE, 0.06*" and the performative is INFORM. The experiment results are demonstrated in Table 3.9 and presented in line diagram shown in Figure 3.27. From the results, it takes more time for transferring messages from the DG modules to the relay modules if the number of both of them increases.

Table 3.9: Time consumption of agent communication in experiment 1.

	The number of DG modules	The number of relay modules					
		5	10	15	20	25	30
Time (ms)	2	94	125	141	218	250	266
	4	125	172	203	265	328	360
	6	140	213	281	328	391	453
	8	157	265	312	375	437	485
	10	172	297	359	422	484	547

Generally, as introduced previously, operating time of a TOC relay is longer than that of a instantaneous overcurrent relay. Normally, it spends hundreds of milliseconds or a couple of seconds being tripped when a fault occurs. From the results of both experiments shown in Tables 3.8 and 3.9, if one message is transferred from a DG module to up to 10 relay modules, the maximum

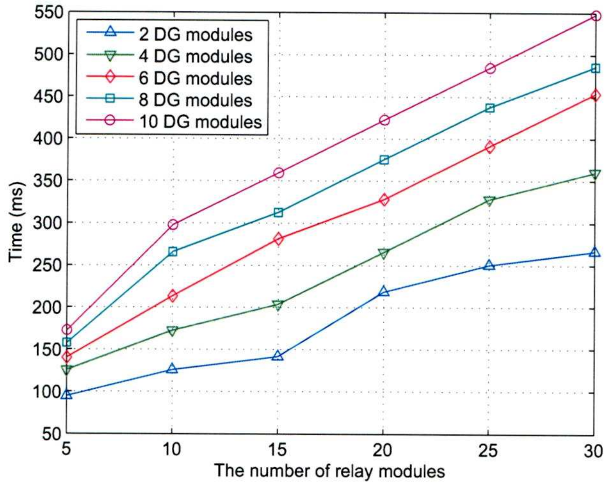


Figure 3.27: Timing performance of agent communication in experiment 2.

time consumption is around 100 ms, which means if a DG unit is connected or disconnected with the existing distribution system, 10 adjacent relay modules can be informed in about 100 ms. Furthermore, it takes about 550 ms for informing 30 relay modules if 10 DG modules are executed at the same time. It is acceptable for coordinating the operations of TOC relays in protecting a distribution network.

3.6 Summary

This chapter has described a relaying scheme developed based on the MAS technology for the improvement of relay coordination in protecting a distribution network with DG integrated. Two types of device modules, *i.e.,* a relay module and a DG module, have been constructed by utilising the specified generic agents that were developed based on the proposed multi-agent framework introduced in Chapter 2. A module is embedded with a device (a protection relay or a DG unit) and communicates with other modules. When a DG unit is connected to the existing distribution network, a DG module is initialised and messages will be sent to its adjacent relay modules. In accordance with the information received from the DG module, a relay module is able to calculate the changes of the measurements affected by the integration of the DG unit and regulate the protection settings of the relay.

In addition, a number of issues for overcurrent protection, including a single-phase-to-ground fault, a three phase-to-ground fault, a phase-to-phase fault, and different situations, such as a single fault issue and a multiple fault issue, have been simulated for evaluating the performance of the proposed scheme. In order to investigate the flexibility and scalability of this scheme, more simulations of two specific situations, such as the disconnection of a DG unit and impedance protection, are also carried out. Furthermore, the timing performance of agent communications in the scheme is discussed.

From the simulation results, The merits of the presented agent-based relaying scheme are in the flexibility, scalability and dynamic response for the protection of a distribution system. It also brings additional value in understanding how DG contributes to the system capacity and the dynamics and operational requirements of a protection relay can be better controlled. In this respect, the chapter offers a clear description of the structure and algorithms governing the behaviour of the constructed relaying scheme.

Chapter 4

Agent Brokering Mechanism for Anti-islanding Protection of Distributed Generation

4.1 Introduction

Distributed generation (DG) has been widely used in the power industry due to market deregulation and environmental concerns. In addition to the protection issues investigated in Chapter 3, such as overcurrent protection and impedance protection, another important requirement of interconnecting DG to distribution systems is the island condition detection of a generator. Islanding operations of DG usually occur when power supply from the main utility is interrupted, but a part of a distribution system containing both loads and operating DG continues to be energised by the DG. The island conditions always cause many serious impacts on protection, operation, and management of distribution systems. For example, a fault may not be cleared since an arc is still fed by DG. Moreover, relay reclosing may couple two asynchronously operating systems, because the frequency of the islanded part of the system may have changed due to unbalanced active power. Therefore, it is necessary to effectively detect the island conditions and disconnect DG from distribution network promptly. The current industry practice is to disconnect all DG units within 100 to 300 ms after loss of main supply [101].

Generally, if there are large changes in loading for DG after loss of the main power supply, then islanding conditions are easily detected by monitoring several parameters: voltage magnitude, phase displacement, and frequency change. However, in case of small changes in loading for DG, the conventional methods have some difficulty in detecting such a particular islanding condition [102]. Furthermore, it is very complicated and expensive for utilising high-speed peer-to-peer communications among protection relays, because it requires specific communication cables and environments, such as optic fiber and microwave, and the protection devices need to be reconfigured if the network topologies are changed.

This chapter presents an agent brokering-based scheme for anti-islanding protection of DG in a distribution system, which is developed based on the multi-agent framework proposed in Chapter 2. The aim of this scheme is to retrieve operational signals from a protection relay and transfer these signals to downstream DG units. Moreover, a command is sent to a DG unit for changing DG connection status in accordance with the received relay operational signals. In this chapter, the impacts of islanding on the protection of DG and the related solutions for anti-islanding protection are introduced. A structure of the proposed agent brokering-based scheme is then presented, followed by a description of the behaviours and the integrated algorithms of each agent employed in this scheme. Furthermore, simulation studies for evaluating the performance of the proposed scheme in the different situations are introduced and a discussion of the simulation results including the timing performance is given.

4.1.1 Impacts of islanding on protection of DG

Basically, islanding is a condition in which a portion of the power networks, containing both loads and operating generations, remains energised when the utility operational procedures require that it be de-energised [103]. As shown in Figure 4.1, a DG unit (DG_1) is connected to the busbar b_2 providing the additional power for the downstream load. If a fault occurs at the point F_4 and a relay R_1 is tripped. Load L_2 is continuously charged, because DG_1 still works in an operational state.

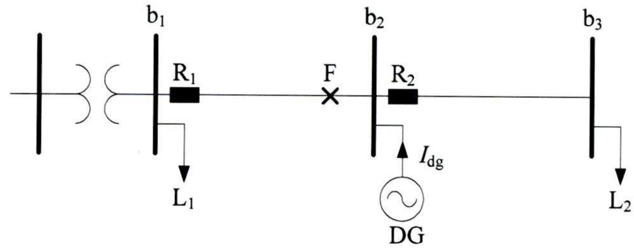


Figure 4.1: Islanding and auto reclosure.

The situation of islanding is generally referred to two types due to the impacts on the protection of DG, such as Loss Of Mains (LOM) and Loss Of Grid (LOG). Generally, islanding may occur due to the following reasons [103]:

- as a result of a fault that is detected by the protective equipments and results in opening a fault interrupting device;
- as a result of accidental opening of the normal utility supply by the equipment failure;
- as a result of utility switching of the distribution system and loads, such as for maintenance operations; or,
- as a result of the human errors or malicious mischiefs.

Additionally, a linear approximation for the frequency change during island operation is given in [104]. Figure 4.2 shows an example for an auto reclosure procedure when a DG unit is not disconnected although it is islanded from the local grid.

As shown in Figure 4.2, the rate of the change of the frequency is expressed as a function of the active power unbalance:

$$\Delta P = \sum P_{dg} - \sum P_l \tag{4.1.1}$$

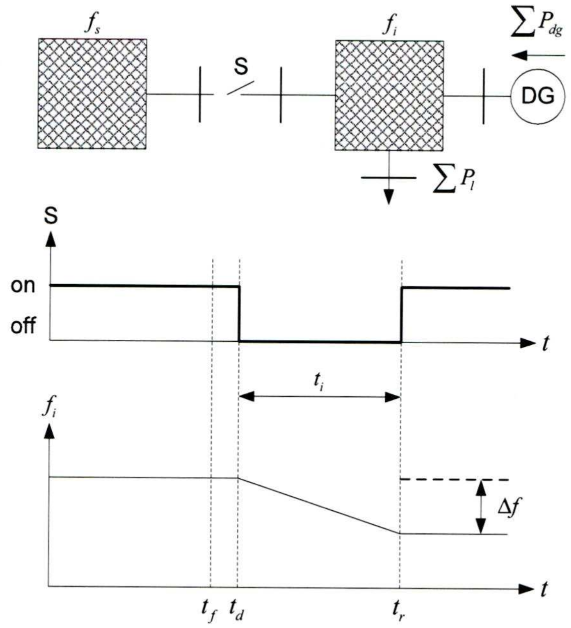


Figure 4.2: Impacts of islanding on frequency.

Furthermore, the inertia constant of the generator H and the system frequency before LOM f_s:

$$\frac{df}{dt} = \frac{\Delta P f_s}{2GH} \tag{4.1.2}$$

where ΔP is the change in active output power during LOM event (MW), as discussed in equation 4.1.1, G is the embedded generator rating (MVA) and H is the inertia constant of the generator (s). Here it is assumed that there is a lack of active power after islanding, *i.e.* $\sum P_{\mathrm{dg}} < \sum P_{\mathrm{l}}$, therefore the island frequency decreases.

In order to limit the outage time of the transient faults, a technique known as "reclosing", is used for opening a fault interrupting device (*i.e.*, a circuit breaker (CB) or a recloser) and then rapidly reclosing it. As described in [4], the common off-time setting of an auto reclosure relay is between 100 ms and 1000 ms, and will be prolonged if DG is integrated with the network. With a continuously operating generator in the network, the following two problems may arise when the utility network is automatically reconnected after a short interruption [4]:

- The fault may not have cleared, since an arc is fed by DG, therefore instantaneous reclosure may not succeed.
- In the islanded part of the network, the frequency may have changed due to the unbalanced active power. Reclosing the breaker will couple two asynchronously operating systems.

4.1.2 Islanding detection methods

The impacts of islanding on the protection of DG have been outlined in Section 4.1. To prevent the island operation, the protection system has to detect the islands quickly and reliably, which is the main objective of the LOM protection. Basically, islanding detection methods can be categorised as the *passive* and the *active* methods [105] [106], which are discussed as follows.

Passive methods

Under/overvoltage Obviously, the voltage is very low if the utility supply is lost. However, it may rise and exceed the maximum limits if there are uncontrolled embedded generators in the power networks. Therefore, the use of the overvoltage relays are regarded as a simple islanding protection method.

Under/overfrequency When LOM occurs, the frequency in the island changes according to equation 4.1.2. Therefore, the islanding operation can be indicated by the frequency exceeding the limits. In this case, the utilisation of the frequency relays is another method for the islanding protection, which is a slow method as the frequency can not be changed instantaneously but continuously.

Voltage vector shift (VVS) The Voltage vector shift (VVS) method, referred to phase displacement [106] or phase jump [107] method, measures the length of each cycle of the voltage wave. As shown in Figure 4.3, part of the load is fed from an embedded generator and the rest is supplied from a utility supply. At the moment the utility supply becomes disconnected, the local load has to be supplied only from the embedded generator and the sudden change in load causes a sudden change in cycle length. Therefore, the transmission angle, *i.e.* the voltage phase difference between the generator and the load terminal rapidly increases due to the sudden power flow increase.

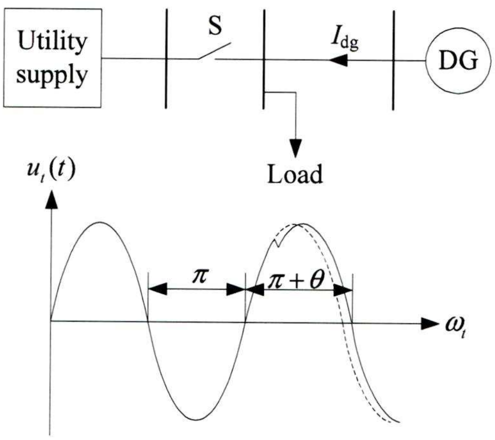


Figure 4.3: Voltage vector shift (VVS).

This increased transmission angle in time domain is shown Figure 4.3 where the dotted lines represent the parallel supply and the solid phasers show the situation after the islanding. Due to the vector jump, the duration of the concerned period is extended. The voltage vector relays monitor the duration of every half cycle and initiate tripping if a certain limit is exceed.

Rate of change of frequency (ROCOF) As discussed previously, the frequency in an island will change rapidly due to the imbalance between the input mechanical power and the prevailing load [108]. Accordingly, the ROCOF is calculated by the following equation:

$$d_{\text{ROCOF}} = \frac{\Delta P \cdot f}{2 \cdot G \cdot H} \tag{4.1.3}$$

Whenever df/dt exceeds a certain limit, the relays are tripped. Typical pickup values are set in a range of 0.1 to 1.0 Hz/s, the operating time is between 0.2 and 0.5 s [92] [98]. The actual calculation of df/dt by a relay can be performed in many alternative ways; many manufacturers use their own specific implementation of the frequency measurement and the processing algorithms. Furthermore, in order to minimise the chance of spurious tripping, it is a routine that there are two or more consecutive calculations which must indicate a setting threshold violation before the trip signal is initiated.

Active islanding detection methods

Apart from the passive islanding detection and protection methods, there are also the active methods being developed to detect the islanding by directly interacting with the system to get an indication of islanding operation. They are listed as follows.

Reactive error export The reactive error export detector controls the embedded generator excitation current so that it generates a known value of reactive current, which cannot be supported unless the generator is connected to the grid [109]. This reactive export error is taken as an indicator for LOM.

Fault level monitoring The fault level monitoring method uses point-on-wave thyristor switching, which triggers when it is close to the voltage zero point, and measures the current through a shunt inductor, enabling rapid calculation (every half cycle) of system impedance and fault level [109].

System impedance monitoring The system impedance monitoring method detects LOM by actively monitoring the system impedance [105]. A high frequency (HF) source is connected at the interconnection point via a coupling capacitor which is in series with the equivalent network impedance. When the systems are synchronised, the impedance is low, therefore the HF-ripple at coupling point is negligible. Once islanding, the impedance increases dramatically and the divided HF-signal is clearly detectable.

Frequency shift Inverter-interfaced DG can be protected against LOM using frequency shift methods [110] [111]. The output current of the converter is controlled by a frequency which is slightly different to the nominal frequency of the system. This is done by varying the power factor during a cycle and re-synchronising at the beginning of a new cycle. Under normal conditions, the terminal frequency is dictated by the powerful bulk supply. If the mains supply is lost, frequency will drift until a certain shutdown level is exceeded.

4.1.3 Related work

Jang [102] presented a new islanding detection algorithm for DG effectively working in most of DG loading conditions. Two new parameters for detecting islanding operation of DG were proposed, *i.e.,* voltage unbalance and total harmonic distortion of current. Moreover, this algorithm was tested using the radial distribution network of IEEE 34-bus model. The test results showed that the proposed parameters and algorithm are capable of correctly detecting the islanding operation are not affected by variation of DG loading and also have a good selectivity for islanding conditions and non-islanding conditions.

Freitas [112] presented a simple and reliable method for predicting the islanding detection performance of vector surge relays. Analytical formulas were developed for directly determining the behaviour of vector surge relays. To cope with practical cases where loads usually exhibit an aggregate voltage-dependent characteristic, an empirical factor is proposed to incorporate power imbalance variations into the analytical formula. An extensive comparison between the formula calculation and simulation results indicated that the proposed formula can predict the performance of vector surge relays with good accuracy. Furthermore, the proposed method can be utilised to avoid time-consuming simulations at the planning and implementation stages of vector surge relays.

González [113] proposed an active islanding detection method for electronically-interfaced distributed resource units at the distribution voltage level. This method is based on injecting a disturbance signal into the system through either the direct axis (d-axis) or the quadrature axis (q-axis) current controllers of the interface voltage-sourced converter. Signal injection through the d-axis controller modulates the amplitude of the voltage at the point of common coupling (PCC), whereas signal injection through the q-axis controller causes a frequency deviation at PCC, under islanded conditions. The simulation studies were carried out based on time-domain simulations in the PSCAD/EMTDC environment. The results showed that this islanding detection method succeeds in detecting the islanding phenomenon as fast as 33.3 ms for the parameter setting of the test system, and always meets the two-second UL detection requirement.

Xu [114] presented an innovative power line signaling based anti-islanding scheme developed in response to the challenge. The scheme broadcasts a signal from a substation to the DG sites using the distribution feeders as the signal paths. Specifically, a signal generator installed at the substation for sending the islanding signals and information over power lines and a signal detector is connected to the DG terminal to examine the presence of the anti-islanding signal and determine if an islanding condition has occurred. A DG unit is considered as islanded from the upstream system if the signal cannot be detected at the DG site. A significant feature of the proposed scheme is that the signal is extracted by subtracting two consecutive voltage cycles. Furthermore, the proposed scheme was evaluated using analytical, simulation and field tests. The results were very promising and indicated that the proposed scheme performed successfully during the field test and the improved algorithms, either spectrum-based or template based, are more reliable for signal detection. In addition, the investigation on the interference between the automatic metre reading (AMR) and anti-islanding schemes showed that the AMR system is unlikely to cause problems to the anti-islanding scheme.

4.2 Development of An Agent Brokering-based Anti-islanding Scheme

This section presents the development of the proposed agent brokering-based scheme for the protection of DG in a distribution network. As shown in

Figure 4.4, this scheme is to transfer relay operating signals (trip signals and reclosure signals) to downstream DG units. Furthermore, the actions of the breaker that control the connection of the DG unit can be coordinated with the operations of the relay. In this section, an overall structure of the proposed scheme is introduced followed by the descriptions of the three developed modules, such as a relay module, a DG module and a broker module.

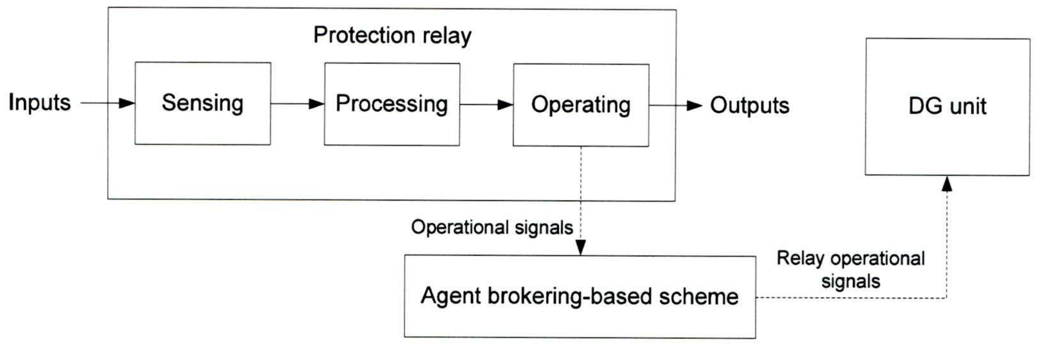


Figure 4.4: An agent brokering-based anti-islanding protection scheme.

4.2.1 A structure of agent brokering-based scheme

The structure of the proposed agent brokering-based anti-islanding protection scheme, shown in Figure 4.5, is composed of three modules: a relay module connected to a protection relay, a DG module installed at the terminal of a DG unit, and a broker module including a number of broker agents for supporting the coordination between these two modules. Both the relay module and the DG module consist of three generic agents, *i.e.,* a control agent ($\mathcal{A}_{\mathrm{CA}}$), a database agent ($\mathcal{A}_{\mathrm{DGA}}$) and a communication agent ($\mathcal{A}_{\mathrm{ComA}}$), which are developed based on the generic agent structure described in Chapter 2.

Particularly, the constructions of the relay and DG modules designed in this scheme are the same as those developed in the agent-based relaying scheme introduced in Chapter 3. However, the functions provided by them are different. In this scheme, the relay module collects the operating signals from a protection relay and requests the broker module to match downstream DG modules

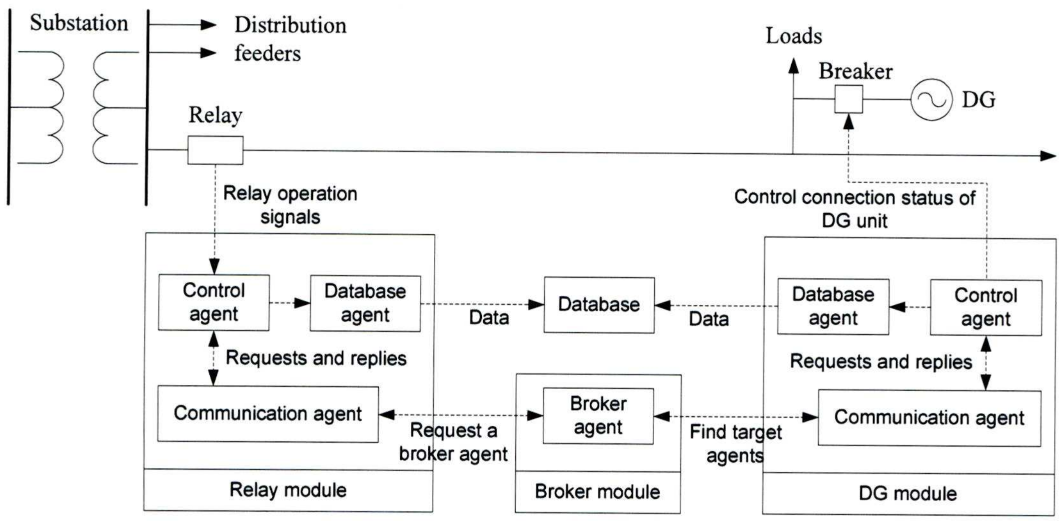


Figure 4.5: A structure of an agent brokering-based scheme.

and forward the signals to them. According to received relay operating signals and the pre-defined control rules, the DG module is capable of controlling the connection status of the DG unit. Furthermore, the DG module is able to determine the status of the upstream network. This feature is particularly useful when two or more faults occur in the system simultaneously, because the DG unit can not be allowed to be reconnected if a relay reclosure signal is received, but the power supplier is still lost.

In addition, the communication infrastructure of the proposed scheme is based on both WAN and LAN. The LAN components represent those modules employed in the scheme that would be installed at a substation, while the WAN components represent those modules found at other locations, such as a wind farm or on a client's computer. Agent communications between the modules are based on the combined FIPA and UDP protocols introduced in Section 2.4.

4.2.2 Relay module

The relay module is defined as $\mathcal{M}_{\mathrm{R}}$, and the generic agents employed in this module are presented as $\mathcal{A}_{\mathrm{CA\_R}}$, $\mathcal{A}_{\mathrm{DBA\_R}}$ and $\mathcal{A}_{\mathrm{ComA\_R}}$, respectively. Particularly, $\mathcal{M}_{\mathrm{R}}$ is expressed as:

$$\mathcal{M}_{\mathrm{R}} \supset \{\mathcal{A}_{\mathrm{CA\_R}}, \mathcal{A}_{\mathrm{DBA\_R}}, \mathcal{A}_{\mathrm{ComA\_R}}\}$$

The capabilities of the relay module are collecting runtime relay operating signals, requesting the broker module and organising the relay operational data in a database, which are introduced as follows.

Receive relay operating signals

$\mathcal{A}_{\mathrm{CA\_R}}$ is able to receive operating signals from it connected relay and define a parameter *Relay_status* to indicate the runtime relay status, which can be expressed as follows:

$$Relay\_status = \begin{cases} \text{OPEN} & \text{if a relay is tripped} \\ \text{CLOSE} & \text{if a relay is reclosed} \end{cases}$$

Once the relay is operated, *Relay_status* is changed and transferred to both $\mathcal{A}_{\mathrm{DBA\_R}}$ and $\mathcal{A}_{\mathrm{ComA\_R}}$ for saving the relay operational data and requesting the broker module to contact the relevant DG modules.

Register service and request the broker module

When the relay module is initialised, an ACL message is sent by $\mathcal{A}_{\mathrm{ComA\_R}}$ to the broker agent to register the information of its location and provided services. The performative of this message is INFORM and the contents include "*ID_ComA_R, b_i*", where *ID_ComA_R* is the identification number of $\mathcal{A}_{\mathrm{ComA\_R}}$ and *b_i* is the relay connected busbar.

In addition, if the connected relay is tripped or reclosed, a request message generated based on the preset message template is sent to the broker module by $\mathcal{A}_{\mathrm{ComA\_R}}$. The contents of the message are "*ID_ComA_R, Relay_status*" and the performative is REQUEST. Moreover, an "AGREE" message is received from the broker module if this request is accepted and forwarded to the related DG modules. Otherwise, if the message can not be recognised or no corresponding DG module is matched, a "REFUSE" or a "FAILURE" message is received.

Record relay operational data

$\mathcal{A}_{\text{DBA\_R}}$ maintains the connection with a database using a preset parameter *DB_address* that is the IP address of the database. When a *Relay_status* parameter is received from $\mathcal{A}_{\text{CA\_R}}$, it is inserted into a column in the database table. The recorded data can be extracted and utilised by engineers for monitoring the overall condition of the distribution system.

4.2.3 DG module

The DG module in the proposed scheme is defined as $\mathcal{M}_{\text{DG}}$. Agents employed in this module are presented as $\mathcal{A}_{\text{CA\_DG}}$, $\mathcal{A}_{\text{DBA\_DG}}$ and $\mathcal{A}_{\text{ComA\_DG}}$, respectively. $\mathcal{M}_{\text{R}}$ is expressed as:

$$\mathcal{M}_{\text{DG}} \supset \{\mathcal{A}_{\text{CA\_DG}}, \mathcal{A}_{\text{DBA\_DG}}, \mathcal{A}_{\text{ComA\_DG}}\}$$

Four functions are provided by this module, such as registering its location and service information in the broker module, receiving requests from the broker module, coordinating the connection status of the embedded DG unit and managing the DG operational data in a database.

Register services and receive requests

When the DG module is initialised, an ACL message is sent by $\mathcal{A}_{\text{ComA\_DG}}$ to the broker module for the registration of its location information. The performative of this message is INFORM and the contents are "*ID_ComA_DG, b_i*", where *ID_ComA_DG* is the identification number of the communication agent employed in the DG module and *b_i* is the DG unit connected busbar.

Additionally, it is able to receive request messages forwarded by the broker module from the relay modules. The contents of this message include "*ID_BA, ID_ComA_R, Relay_status*", where *ID_BA* is the identification number of the broker agent and the performative of this message is REQUEST. The received requests are transferred to $\mathcal{A}_{\text{CA\_DG}}$ for coordinating the connection status of the DG unit.

Control connection status of a DG unit

In order to meet the requirements of anti-islanding protection, all downstream DG units must be disconnected from the distribution system if the power supplier is lost. In this case, one of the most important issues of this module is to control the connection status of the DG unit according to operations of the upstream relays. The control principles of $\mathcal{A}_{\text{CA\_DG}}$ are to estimate the upstream network status and coordinate the operations of the DG unit. Furthermore, a parameter *DG_status* is defined by $\mathcal{A}_{\text{CA\_DG}}$ based on the collected DG connection signals to indicate the real-time DG connection status, which is presented as follows:

$$DG\_status = \begin{cases} \text{OPEN} & \text{if a DG unit is disconnected} \\ \text{CLOSE} & \text{if a DG unit is reconnected} \end{cases}$$

In accordance with the requests received by $\mathcal{A}_{\text{ComA\_DG}}$, another parameter *NW_status* presenting the upstream network status is defined by $\mathcal{A}_{\text{CA\_DG}}$ following Algorithm 3.

Algorithm 3 Determination of *NW_status*

Begin

If	$Relay1\_status \cap Relay2\_status \cap ... \cap Relayn\_status = \text{CLOSE}$
Then	$NW\_status = \text{CLOSE}$
Else if	$Relay1\_status \cup Relay2\_status \cup ... \cup Relayn\_status = \text{OPEN}$
Then	$NW\_status = \text{OPEN}$

End

As introduced in Section 4.1, a DG unit must be disconnected if one of its upstream relay is tripped due to a fault in a distribution network. Furthermore, if more than one relays located in the upstream network of the DG unit are tripped, the DG unit can not be reconnected until all of the upstream relay are reclosed. In accordance with this principle, the control rules built in $\mathcal{A}_{\text{CA\_DG}}$ are listed as follows:

If the DG unit is connected and the upstream network is cut off, $\mathcal{A}_{\text{CA\_DG}}$ sends an "OPEN" command to trip the DG unit, otherwise, if the DG unit is

Algorithm 4 Send a command to the DG unit

Input

DG_status: connection status of the DG unit

NW_status : upstream network status

Begin

If DG_status = CLOSE **AND** NW_status = OPEN

Then $SendCommand(OPEN)$

Else if DG_status = OPEN **AND** NW_status = CLOSE

Then $SendCommand(CLOSE)$

End

disconnected and all the upstream relays are reclosed, a "CLOSE" command is sent.

4.2.4 Broker module

The broker module is composed of a number of broker agents for coordinating the relay modules and the DG modules in the proposed scheme. A broker agent employed in this module is defined as $\mathcal{A}_{\mathrm{BA}}$, which is developed based on the agent-brokering mechanism introduced in Section 2.5.2. A knowledge base is managed by the broker agent for maintaining the information of the distribution network topologies and the registered services of each executed module. The main responsibility of the broker agent is to receive requests from relay modules, match the related DG modules in the knowledge base and forward the requests to them. In particular, three collaborative behaviours are carried out by $\mathcal{A}_{\mathrm{BA}}$. For example, a *RegisterBehaviour* is responsible for agent service registration. A *KnowledgeBehaviour* is able to manage the knowledge base. A request message received from the relay module is processed and forwarded to the matched tagger DG modules by a *RequestBehaviour*.

RegisterBehaviour

A number of registration messages are received from both $\mathcal{A}_{\mathrm{ComA\_DG}}$ and $\mathcal{A}_{\mathrm{ComA\_R}}$ when the are initialised. The performative of each message is IN-

FORM and the contents include "*ID_ComA_DG/ID_ComA_R, Connected_busbar*", where *ID_ComA_DG* and *ID_ComA_R* are the identification numbers of $\mathcal{A}_{\text{ComA\_DG}}$ and $\mathcal{A}_{\text{ComA\_R}}$ respectively. If the message is accepted, an "ACCEPT" message is replied to the sender and it is forwarded to the *KnowledgeBehaviour* for adding the provided services into the knowledge base. Otherwise, a "REJECT" message is replied if the request is rejected or the message can not be recognised.

KnowledgeBehaviour

The *KnowledgeBehaviour* is responsible for maintaining the knowledge base to match the relevant DG modules according to the requests received from the relay modules. The knowledge base stores the topology information of the distribution system and relationships between the relay and DG modules. Specifically, the network topology information is expressed as:

$$b\_1 \supseteq \{b\_2, b\_3......b\_n\}$$

where b_1, b_2 and b_n present the busbar b_1, b_2 and b_n, respectively. This expression means that b_1 situates in the upstream network of the busbars b_2, b_3 and b_n. Using the registration messages received from $\mathcal{A}_{\text{ComA\_DG}}$ and $\mathcal{A}_{\text{ComA\_R}}$, the location information of the relay and DG modules can be defined as:

$$\mathcal{A}_{\text{ComA\_R}}(ID\_ComA\_R, b\_i)$$
$$\mathcal{A}_{\text{ComA\_DG}}(ID\_ComA\_DG, b\_j)$$

Based on the network topologies, a list including the relationships between a relay module and its downstream DG modules is generated. For example, if the relay R_3 is located in the upstream network of the DG units DG_2, DG_3, and DG_4, the relationship can be expressed as:

$$\mathcal{A}_{\text{ComA\_R3}} \supseteq \{\mathcal{A}_{\text{ComA\_DG2}}, \mathcal{A}_{\text{ComA\_DG3}}, \mathcal{A}_{\text{ComA\_DG4}}\}$$

In addition, the *KnowledgeBehaviour* is capable of matching the target DG modules in the knowledge base and generating a sender table presented as $Sender(\mathcal{A}_{\text{ComA\_DGi}})$ which will be forwarded to the *RequestBehaviour*. Furthermore, if the locations of the relay and DG modules are changed, or new relay and DG modules are emerged into the system, this list can updated automatically.

RequestBehaviour

This behaviour is to handle a request from a relay module and forward it to the related DG modules. A message including "*ID_ComA_R, Relay_status*" is received from $\mathcal{A}_{\text{ComA\_R}}$, and the performative is REQUEST. In accordance with the sender table generated by the *KnowledgeBehaviour*, this request is forwarded to all of the DG modules appeared in $Sender(\mathcal{A}_{\text{ComA\_DGi}})$. Specifically, the contents of this message include "*ID_BA, ID_ComA_R, Relay_status*" and the performative is REQUEST. Furthermore, reply messages received from the DG modules are delivered to $\mathcal{A}_{\text{ComA\_R}}$.

4.3 Simulation System

4.3.1 A simulation model of a distribution network

A distribution network model is built in the PSCAD/EMTDC programme environment, shown in Figure 4.6. Four DG units are integrated with the network through the busbars b_2, b_3, b_6 and b_8, respectively. In this network, the utilised power supplier is configured as a voltage source with 100 MVA output power and 66 kV constant output voltage. The impedance of the voltage source is 5 Ω and the phase angle is 80°. Each DG unit is comprised of four individual wind power generators providing 8 MVA output power and a three-phase constant current. The protection relays installed in the system are configured as time-overcurrent relays that follow IEEE Extremely Inverse TOC Curves. The operating time of breakers and reclosers utilised in this model are configured as 20 ms. Furthermore, the distances between the busbars in the distribution system are listed in Table 4.1

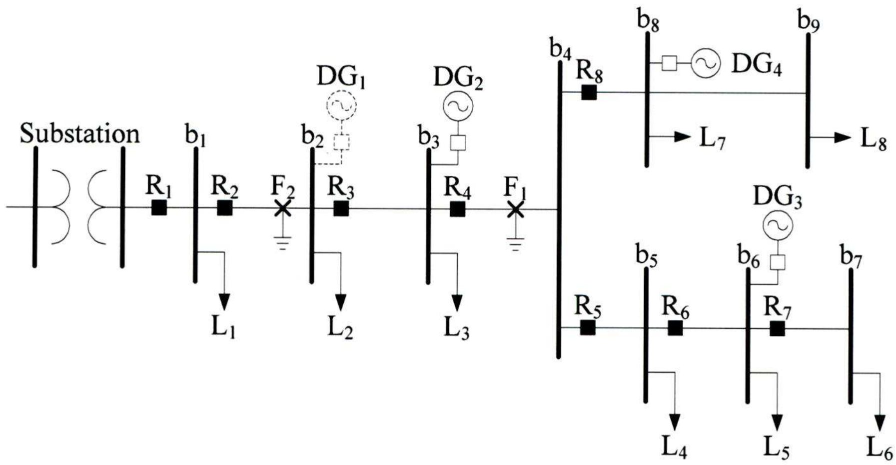


Figure 4.6: The distribution network model for simulation study.

Table 4.1: Distance between the busbars.

Busbar	Distance	Busbar	Distance
b_1 to b_2	10 km	b_5 to b_6	10 km
b_2 to b_3	15 km	b_6 to b_7	5 km
b_3 to b_4	15 km	b_4 to b_8	15 km
b_4 to b_5	5 km	b_8 to b_9	10 km

4.3.2 Agent communication environment

In order to investigate the performance of agent coordination in the proposed scheme, an agent communication network model developed in the agent-based relaying scheme described in Section 3.3.2 is utilised in this scheme. As shown in Figure 4.7, the protection relays and the DG units are designed using PXI computers that support real-time data acquisition and communication. Both relay and DG modules are executed for communicating with the relays and DG units. A PC is utilised for running the agent platform and the broker module. Each computer connects to the LAN through separate 10/100 Mbps Ethernet switches. In particular, the bandwidth of the Ethernet-based network is 100 Mb/s, and the communication protocols are based on the combined FIPA and UDP protocols introduced in Section 2.4. Data utilised by both the relay and DG modules is generated from each node of the distribution network and saved in a database that is also executed in the agent server computer.

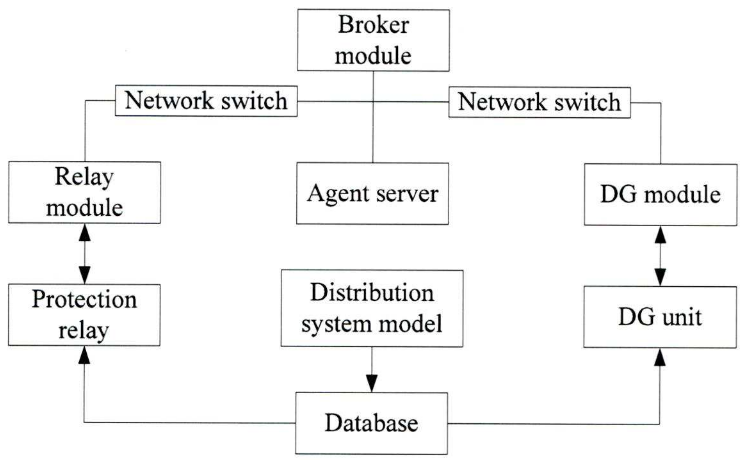


Figure 4.7: The agent communication network.

4.3.3 Simulation scenarios

Two simulation scenarios are carried out to evaluate the performance of the proposed scheme in handling the protection issues of multiple faults and the change of network topologies, respectively.

Scenario I

This scenario aims at investigating the performance of the proposed scheme when multiple faults occur in the distribution system. As shown in Figure 4.6, two single-phase-to-ground faults are set to appear at the points F_1 and F_2 respectively at different times. Particularly, the first fault occurs at 1 s, which is 5 km away from b_3, with 30 Ω fault resistance and the fault arc extinguish time is set as 200 ms. R_4 is tripped at 1.090 s and set to be reclosed at 1.590 s. The second fault occurs at 1.2 s, which is 10 km away from b_2, with 10 Ω fault resistance. Following its characteristics, R_2 is tripped at 1.260 s and reclosed at 1.760 s.

Scenario II

In the second scenario, the flexibility of the proposed scheme is evaluated that a new DG unit is interconnected into the existing system in runtime. As displayed in Figure 4.6, DG_1 (drawn in dashed line) is to be connected to the busbar b_2 at 0.5 s and a three-phase-to-ground fault occurs at F_2 at 1.2 s, which is 10 km away from b_2, with 5 Ω fault resistance. R_2 is tripped at 1.240 s and reclosed at 1.740 s.

4.4 Simulation Results

4.4.1 Multiple faults scenario

At the initialised stage, eight relay modules ($\mathcal{M}_{\mathrm{R1}}$, $\mathcal{M}_{\mathrm{R2}}$, *etc.*) and four DG modules ($\mathcal{M}_{\mathrm{DG1}}$, $\mathcal{M}_{\mathrm{DG2}}$, *etc.*) are executed and register their location information in the broker agent. Following the first simulation scenario described in Section 4.3.3, when R_4 is tripped at 1.090 s, the trip signal is collected by $\mathcal{A}_{\mathrm{CA\_R4}}$. The parameter *R4_status* indicating current status of R_4 is then set as "OPEN".

As shown in Figure 4.8, *R4_status* is sent to $\mathcal{A}_{\mathrm{BA}}$ by $\mathcal{A}_{\mathrm{CA\_R4}}$. Then, $\mathcal{A}_{\mathrm{BA}}$ matches two downstream DG modules ($\mathcal{M}_{\mathrm{DG3}}$ and $\mathcal{M}_{\mathrm{DG4}}$) and forwards the request to them. At 1.180 s, both $\mathcal{A}_{\mathrm{ComA\_DG3}}$ and $\mathcal{A}_{\mathrm{ComA\_DG4}}$ receive a request message and set the network status parameters *NW3_status* and *NW4_status* as "OPEN" following Algorithm 3.

In accordance with Algorithm 4, an "OPEN" command is sent to DG_3 and DG_4 at 1.181 s and 1.183 s, respectively. Accordingly, DG_3 and DG_4 are then disconnected. Figure 4.9(a) illustrates the measured waveform of phase-A current and the trip signals of R_4. Moreover, the current waveform and the connection signals detected by DG_3 and DG_4 are displayed in Figures 4.10(a) and 4.10(b), respectively.

When the second fault occurs and R_2 is tripped, the operational signal is collected by $\mathcal{A}_{\mathrm{CA\_R2}}$ and the parameter *R2_status* is set as "OPEN". A request

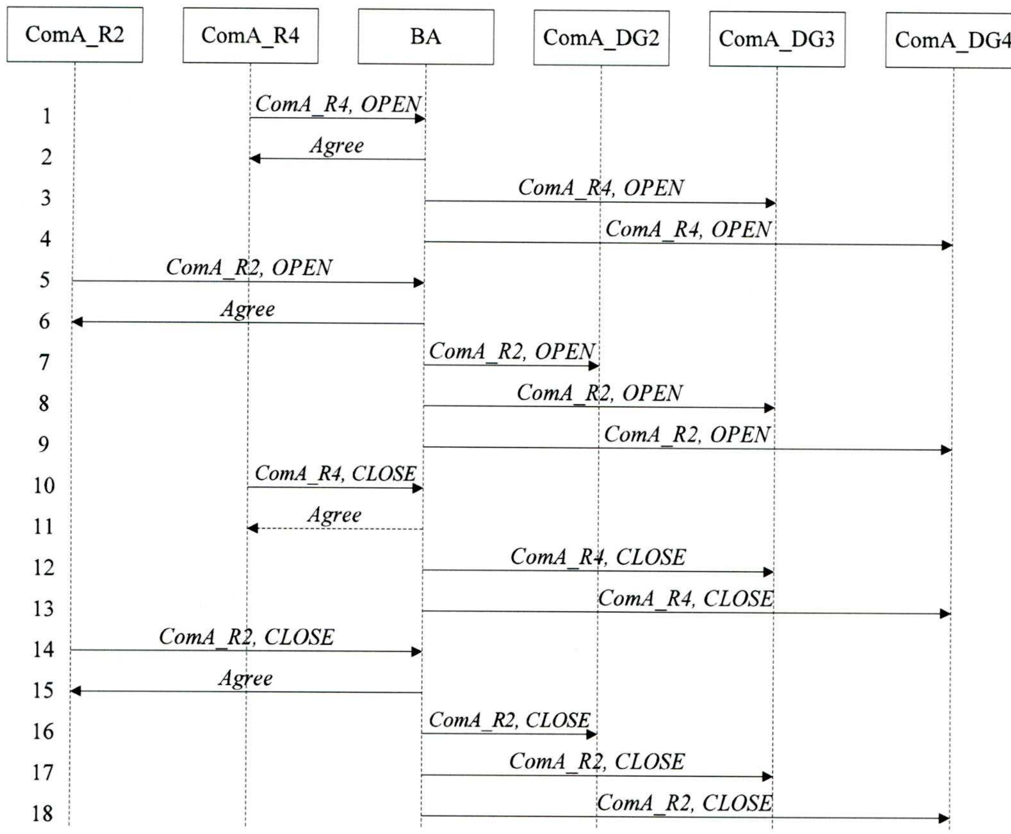
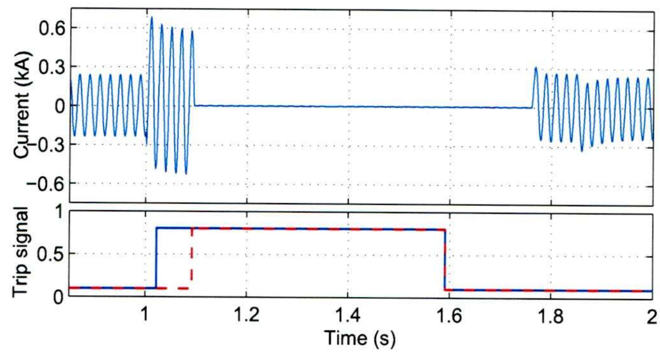


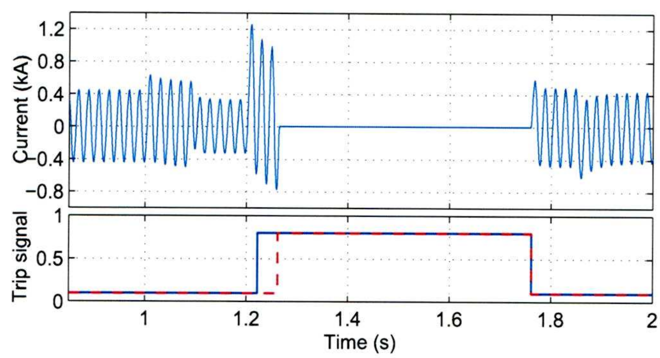
Figure 4.8: Agent coordination in simulation scenario I of the agent-brokering based scheme.

message including *R2_status* is sent to $\mathcal{A}_{\text{BA}}$ that matches three downstream DG modules ($\mathcal{M}_{\text{DG2}}$, $\mathcal{M}_{\text{DG3}}$ and $\mathcal{M}_{\text{DG4}}$) and forwards this request to them. $\mathcal{A}_{\text{ComA\_DG2}}$ receives this request at 1.350 s and *NW2_status* is set as "OPEN" in accordance with Algorithm 3. An "OPEN" command is then sent to trip DG_2 immediately following Algorithm 4. However, no actions are taken by both $\mathcal{A}_{\text{ComA\_DG3}}$ and $\mathcal{A}_{\text{ComA\_DG4}}$, because both *DG3_status* and *DG4_status* are set as "OPEN" currently. The operations and time stamps taken by each agent in this process are displayed in Table 4.2.

With the same communication process, when R_4 is set to be reclosed at 1.590 s, *R4_status* is set as "CLOSE". A request message is sent to $\mathcal{A}_{\text{BA}}$ and forwarded to $\mathcal{A}_{\text{ComA\_DG3}}$ and $\mathcal{A}_{\text{ComA\_DG4}}$ respectively. However, *NW3_status* and *NW4_status* are still set as "OPEN" since *R2_status* is "OPEN". Therefore,



(a) Measurements and operational signals of R_4.

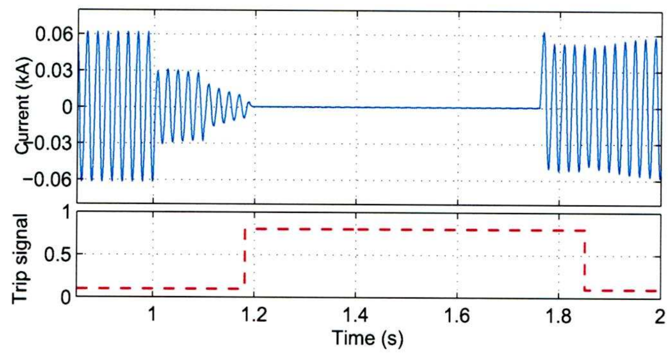


(b) Measurements and trip signals of R_2.

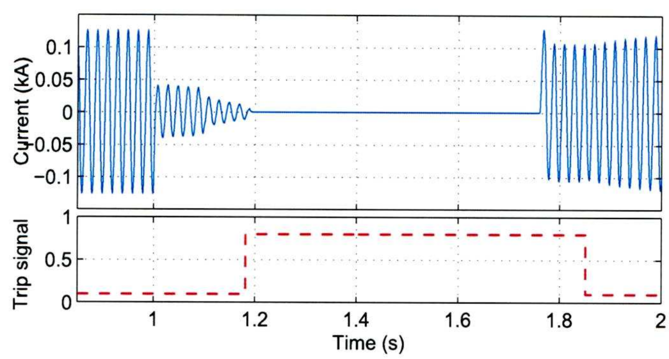
Figure 4.9: Relay measurements and operation signals in scenario I of agent brokering-based scheme.

$\mathcal{A}_{\mathrm{CA\_DG3}}$ and $\mathcal{A}_{\mathrm{CA\_DG4}}$ keep DG_3 and DG_4 being disconnected. As displayed in Figs. 4.10(a) and 4.10(b), after 1.590 s there is still no current measured by both DG_3 and DG_4.

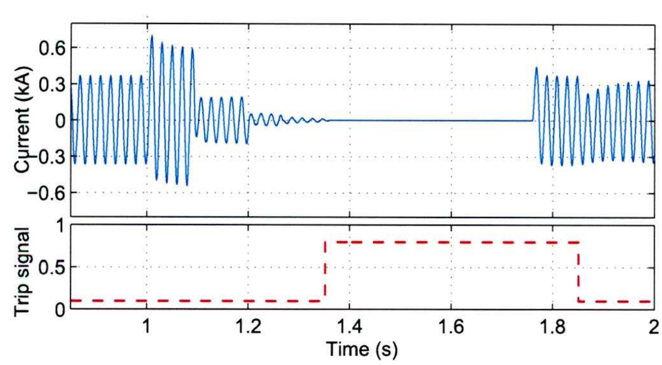
At 1.760 s R_2 is reclosed, shown in Figure 4.9(b), $\mathcal{A}_{\mathrm{CA\_R2}}$ collects a "CLOSE" signal and changes *R2_status* to "CLOSE". At the same time, $\mathcal{A}_{\mathrm{ComA\_R2}}$ sends a message to request $\mathcal{A}_{\mathrm{BA}}$ to contact the downstream DG modules for reconnecting the DG units. As indicated in Table 4.2, a "CLOSE" command is sent by $\mathcal{A}_{\mathrm{ComA\_DG2}}$, $\mathcal{A}_{\mathrm{ComA\_DG3}}$ and $\mathcal{A}_{\mathrm{ComA\_DG4}}$ at 1.852 s, 1.853 s, and 1.856 s respectively to reconnect DG_2, DG_3 and DG_4.



(a) Current measurements and actions of DG_3.



(b) Current measurements and actions of DG_4.



(c) Current measurements and actions of DG_2.

Figure 4.10: Current measurements and actions of DG units in scenario I of agent brokering-based scheme.

4.4.2 Interconnection of a new DG unit

In the second simulation scenario, three DG units ($\mathcal{M}_{DG2}$, $\mathcal{M}_{DG3}$ and $\mathcal{M}_{DG4}$) are integrated with the network at the beginning. Another DG unit (DG_1) is

Table 4.2: Time stamps of agent operations in scenario I of agent brokering-based scheme.

Time (s)	Agent ID	Operation	Parameter
1.090	$\mathcal{A}_{\mathrm{CA\_R4}}$	R_4 tripped	*R4_status* = OPEN
1.181	$\mathcal{A}_{\mathrm{ComA\_DG3}}$	send a command	“OPEN”
1.181	$\mathcal{A}_{\mathrm{CA\_DG3}}$	DG_3 tripped	*DG3_status* = OPEN
1.183	$\mathcal{A}_{\mathrm{ComA\_DG4}}$	send a command	"OPEN”
1.183	$\mathcal{A}_{\mathrm{CA\_DG4}}$	DG_4 tripped	*DG4_status* = OPEN
1.260	$\mathcal{A}_{\mathrm{CA\_R2}}$	R_2 tripped	*R2_status* = OPEN
1.350	$\mathcal{A}_{\mathrm{ComA\_DG2}}$	send a command	“OPEN”
1.350	$\mathcal{A}_{\mathrm{CA\_DG2}}$	DG_2 tripped	*DG2_status* = OPEN
1.590	$\mathcal{A}_{\mathrm{CA\_R4}}$	R_4 reclosed	*R4_status* = CLOSE
1.760	$\mathcal{A}_{\mathrm{CA\_R2}}$	R_2 reclosed	*R2_status* = CLOSE
1.852	$\mathcal{A}_{\mathrm{ComA\_DG2}}$	send a command	“CLOSE”
1.852	$\mathcal{A}_{\mathrm{CA\_DG2}}$	DG_2 connected	*DG2_status* = CLOSE
1.853	$\mathcal{A}_{\mathrm{ComA\_DG3}}$	send a command	“CLOSE”
1.853	$\mathcal{A}_{\mathrm{CA\_DG3}}$	DG_3 connected	*DG3_status* = CLOSE
1.856	$\mathcal{A}_{\mathrm{ComA\_DG4}}$	send a command	"CLOSE”
1.856	$\mathcal{A}_{\mathrm{CA\_DG4}}$	DG_4 connected	*DG4_status* = CLOSE

planned to be connected to b_2 at 0.5 s after the simulation is started. As introduced in Section 4.2.3, the DG module ($\mathcal{M}_{\mathrm{DG1}}$) registers its local information with $\mathcal{A}_{\mathrm{BA}}$ immediately when it is executed.

As displayed in Figure 4.11, a message including "*ComA_DG1, b2*" is sent over by $\mathcal{A}_{\mathrm{ComA\_DG1}}$. According to this message, $\mathcal{A}_{\mathrm{BA}}$ updates its knowledge base that $\mathcal{A}_{\mathrm{ComA\_DG1}}$ is related to $\mathcal{A}_{\mathrm{ComA\_R2}}$ as one of its downstream targets. The time consumption of the registration is approximately 40 ms. As illustrated in Figure 4.12(a), DG_1 is interconnected with the system at 0.5 s and R_2 is tripped at 1.240 s due to a fault occurring at 1.20 s. The trip signal is sent by $\mathcal{A}_{\mathrm{ComA\_R2}}$ immediately and is forwarded to the downstream DG modules by $\mathcal{A}_{\mathrm{BA}}$. Particularly, $\mathcal{A}_{\mathrm{ComA\_DG1}}$ is emerged into the existing system and matched by $\mathcal{A}_{\mathrm{BA}}$, and it is able to receive the request message to coordinate

DG operation.

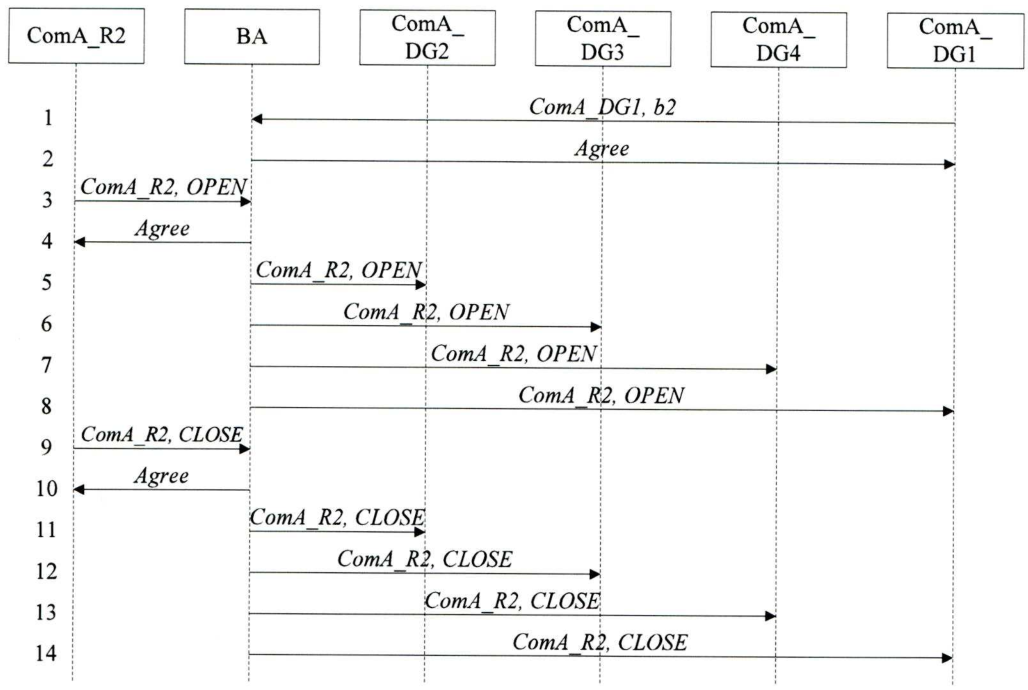
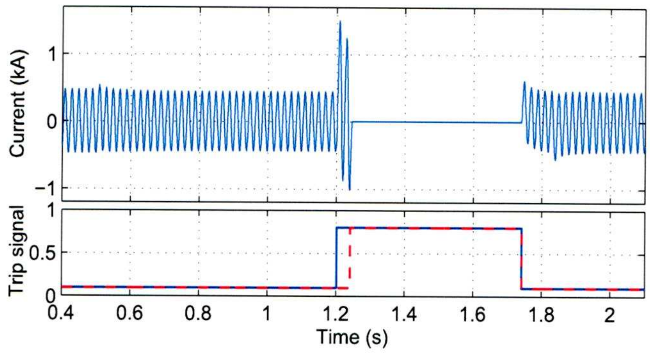


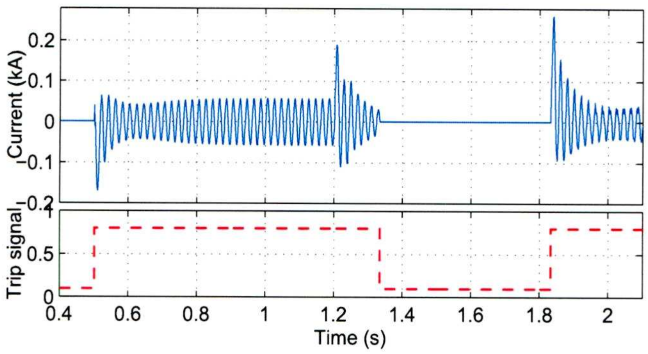
Figure 4.11: Agent coordination in simulation scenario II of the agent-brokering based scheme.

Table 4.3: Time stamps of agent operations in scenario II of agent brokering-based scheme.

Time (s)	Agent ID	Operation	Parameter
0.500	$\mathcal{A}_{\text{ComA\_DG1}}$	send a message	"ComA_DG1, b2"
1.240	$\mathcal{A}_{\text{CA\_R2}}$	R_2 tripped	*R2_status* = OPEN
1.333	$\mathcal{A}_{\text{ComA\_DG1}}$	send a command	"OPEN"
1.333	$\mathcal{A}_{\text{CA\_DG1}}$	DG_1 tripped	*DG1_status* = OPEN
1.335	$\mathcal{A}_{\text{ComA\_DG2}}$	send a command	"OPEN"
1.335	$\mathcal{A}_{\text{CA\_DG2}}$	DG_2 tripped	*DG2_status* = OPEN
1.336	$\mathcal{A}_{\text{ComA\_DG3}}$	send a command	"OPEN"
1.336	$\mathcal{A}_{\text{CA\_DG3}}$	DG_3 connected	*DG3_status* = OPEN
1.338	$\mathcal{A}_{\text{ComA\_DG4}}$	send a command	"OPEN"
1.338	$\mathcal{A}_{\text{CA\_DG4}}$	DG_4 connected	*DG4_status* = OPEN



(a) Measurements and operational signals of R_4.



(b) Current measurements and actions of DG_1.

Figure 4.12: Current measurements and actions of R_4 and DG_1 in scenario II of agent brokering-based scheme.

Figure 4.12(b) shows the measured current waveform and the operations of DG_1. All the four DG modules receive this request and send an "OPEN" signal to trip the associated DG units. Table 4.3 displays the time stamps of agent operations in this case. After 500 ms, R_2 is reclosed and a "CLOSE" signal is received by $\mathcal{A}_{CA\_R2}$ and forwarded to all of the downstream DG modules. In this case, the DG units are reconnected to the system immediately. The simulation results indicates the proposed anti-islanding scheme is capable of handling the changes of system topology and reconfiguring each agent-based modules integrated with the protection relays automatically.

4.5 Timing Performance of Agent Brokering-based Scheme

From the simulations introduced in the previous section, the flexibility and scalability of the proposed agent brokering-based scheme in anti-islanding protection is investigated. Another important issue for the evaluation of the proposed scheme is the timing performance of the agent communications, which is discussed in this section.

4.5.1 Performance model for broker agent communications

Based on the agent communication environment introduced in Section 4.3.2, an experimental system is established for evaluating the performance of the proposed agent brokering-based scheme in terms of timing of agent communications. Particularly, the timing performance in the experiments is defined as the time consumption of a message being sent from a relay module and received by a DG module. As shown in Figure 4.13, two computers are utilised for the execution of relay modules and DG modules, respectively, and an agent server computer is used for running both the JADE platform and the broker module. All of the computers are configured as Pentium IV 2.8 GHz and connected to the network through the same 100BaseT LAN through network switches. The protocol used for these interactions was based on UDP/IP.

In addition, the performance model proposed in Section 2.5 for the evaluation of the developed agent-brokering mechanism is utilised in this experiment. Specifically, the total time consumption of one message transferred from a relay module to a DG module is defined as:

$$\begin{aligned} T_{\mathrm{msg}} = \; & T_{\mathrm{rm}}(msg, \mathcal{A}_{\mathrm{ComA\_R}}, \mathcal{A}_{\mathrm{BA}}) \\ + \; & T_{\mathrm{bm}}(msg, \mathcal{A}_{\mathrm{BA}}, \mathcal{A}_{\mathrm{ComA\_DG}}) \\ + \; & \delta(\mathcal{A}_{\mathrm{BA}})) \end{aligned} \tag{4.5.1}$$

where T_{rm} and T_{bm} are represented as the time costs of a message delivered

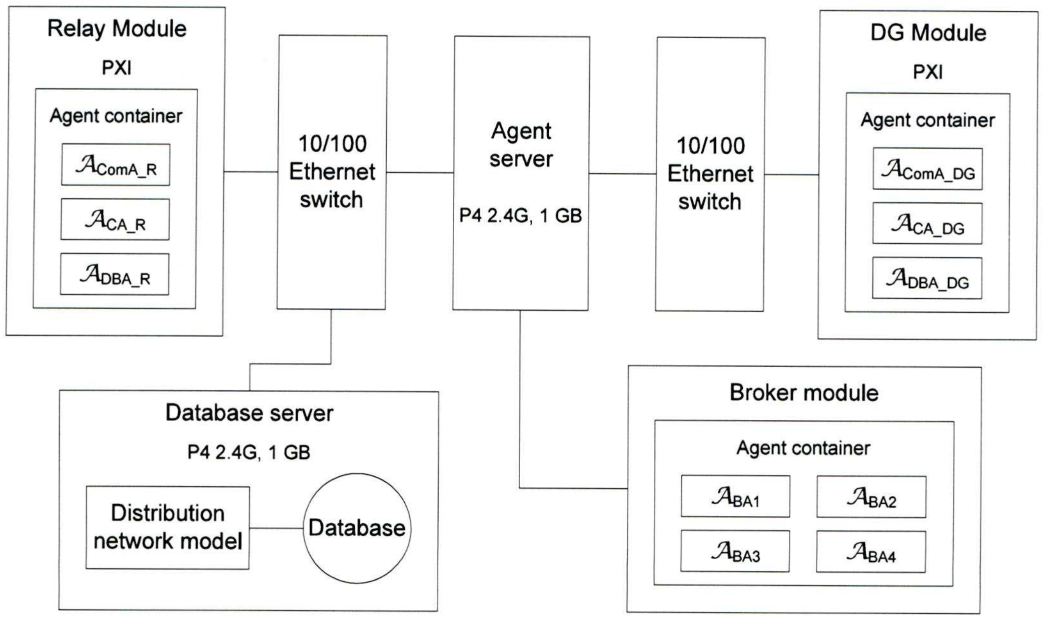


Figure 4.13: Experimental system for evaluating agent brokering-based scheme.

from a relay module to a broker module and from a broker module to a DG module, respectively. $\delta(\mathcal{A}_{\text{BA}})$ is the time latency of $\mathcal{A}_{\text{BA}}$ for matching the relevant $\mathcal{A}_{\text{ComA\_DG}}$ in its knowledge base.

Based on this model, two experiment scenarios are carried out to investigate the response time of the broker module in handling different issues, which are listed as follows:

- **Scenario I:** only one broker agent is employed in the broker module for receiving the request messages from a number of relay modules and forwarding these requests to the relevant DG modules.

- **Scenario II:** the broker module consists of four broker agents for transferring the requests received from the relay modules to the DG modules.

In the following subsections, the detailed descriptions of each scenario are given and the experiment results are discussed.

4.5.2 Evaluation of single brokering scenario

Description

The first scenario aims at investigating the response time of the broker agent with only one broker agent employed for transferring the relay operational signals to the DG modules in the proposed scheme. As described in Section 4.2, the message received from a relay module includes ""*ID_ComA_R, Relay_status*", and the performative of this message is REQUEST.

Particularly, in this scenario, a trip signal is received and *Relay_status* is "OPEN". The number of the relay modules for requesting the broker module is varied from 1 to 5 and the number of employed DG modules is changed from 5 to 20 in increments of 5 each time. Particularly, in order to compare the response time of the broker agent in the different situations, it assumes that all of the relay modules send requests at the same time.

Results

Table 4.4 demonstrates the experiment results of this scenario. Obviously, the time cost increases along with the increments of the transferred messages. For instance, it takes about 156 ms if only one relay module requests the broker agent and this request is forwarded to 10 DG modules, while it costs 406 ms if the requests are received from three relay modules and forwarded to 20 DG modules by the broker agent. The comparison of the experiment results is shown in Figure 4.14.

From the result, the response time of the broker agent employed in the proposed scheme for coordinating the operations between the protection relays and the DG units is acceptable if there are not too many relays operated at the same time. However, if a large number of the relay modules request the broker agent at the same time, the response time of the broker agent significantly increases, since one broker agent can only deal with one task at a time. In this case, if too many requests are received by a broker agent at the same time, it may take quite a long time to match the related DG modules and forward

Table 4.4: Time consumptions of agent communications with one broker agent employed in agent brokering-based scheme.

	Relay module	DG module			
		5	10	15	20
Time (ms)	1	125	156	187	202
	2	203	234	270	312
	3	243	282	335	406
	4	281	328	391	469
	5	1078	1079	1063	1141

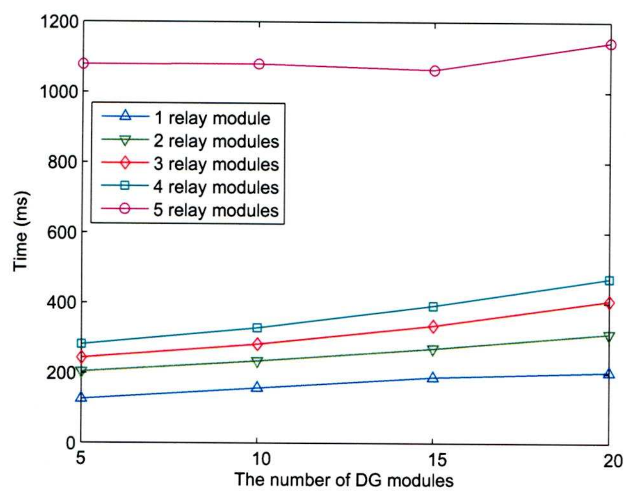


Figure 4.14: Comparison of broker agent response time in scenario I.

these requests to them. Consequently, the operational signals received from the relay modules may not be transferred to the relevant DG modules promptly. Accordingly, the proposed scheme may face bottleneck in this situation and the islanded DG units may not be disconnected immediately.

4.5.3 Investigation on multi-brokering scenario

Multi-brokering mechanism

In order to solve the problem mentioned above, a multi-brokering mechanism is developed based on the agent-brokering mechanism introduced in Chapter 2. The multi-brokering mechanism allows the process of matching service provider agents to requests to be distributed across multiple brokers, which share the same knowledge base. As shown in Figure 4.15, four broker

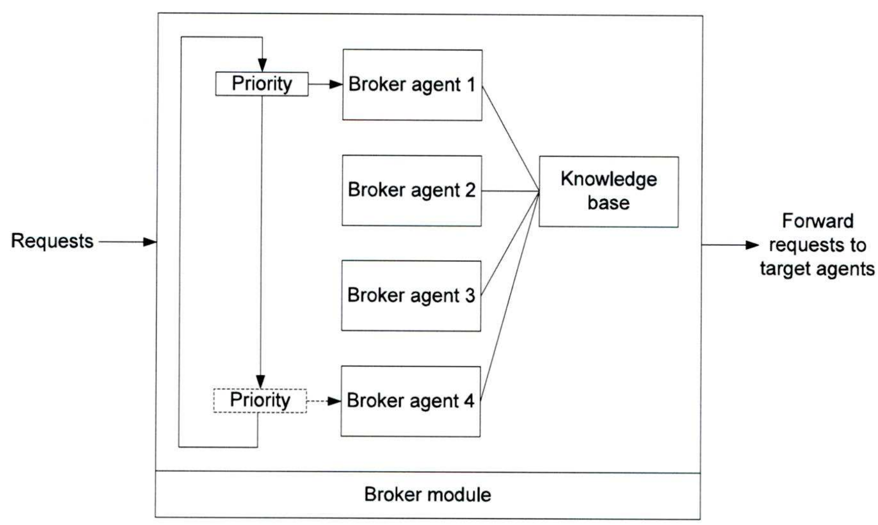


Figure 4.15: A multi-brokering mechanism developed for agent brokering-based scheme.

agents are employed in the broker module. In order to coordinate their operations, a priority list is created and maintained by each broker agent. When the first broker receives a request, it passes the priority to the second broker and forward this request to the relevant DG modules matched in the knowledge base. Additionally, when the fourth broker receives a request, it passes the priority back to the first one.

Results

In this experiment, 10 DG modules are utilised for receiving the requests from the broker module, while the number of the relay modules is varied from

4 to 16 in increments of 4 each time. The response times of the broker module with four broker agents employed in different situations are shown in Table 4.5.

Table 4.5: Time consumptions of agent communications with multiple brokers employed in the agent brokering-based scheme.

	Broker Agents	Relay modules			
		4	8	12	16
Time (ms)	1	344	1329	2297	3235
	2	368	572	1578	2500
	3	391	663	1365	2316
	4	422	656	1244	1875

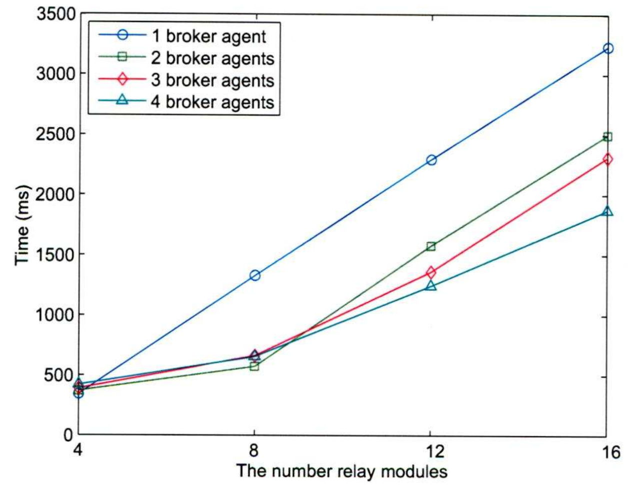


Figure 4.16: Comparison of broker agent response time in scenario II.

Comparisons of the time consumptions by using different numbers of broker agents are illustrated in Figure 4.16. As seen from this figure, if only a few of the relay modules request the broker module, the advantages of the multi-brokering mechanism can not be exhibited, since the number of received messages is not over the maximum capacity of one broker agent and it takes extra time in passing the priority from one broker to another. However, if the executed relay modules increases, the response time of the broker module becomes extraordinary long if only one broker agent is used for handling a large number of requests. In this case, the use of multi-brokering mechanism can remarkably reduce the time costs of transferring messages from a large number of relay modules to the DG modules. For example, if 16 relay modules send request messages at the same time, it costs approximately 3235 ms in handling these requests if only one broker is used, while if the multi-brokering mechanism is applied, the communication time decreases to 1875 ms.

4.6 Summary

This chapter has presented an agent brokering-based scheme developed based on the multi-agent framework for anti-islanding protection of DG. Using the agent brokering mechanism applied in the broker module, it is able to coordinate the operations of relays and the relevant DG units. The proposed scheme is the first attempt for anti-islanding protection using the MAS technology, which is similar to the telecommunication-based protective relaying scheme. However, it does not require special devices or communication cables, therefore it provides a scalable and dynamical option for protecting distribution networks.

Within this scheme, three modules are employed, including a relay module, a DG module and a broker module. The relay module connected with a protection relay is able to collect relay operating signals in real-time and request the broker module to transfer these signals to its downstream DG modules. According to the received relay operating signals, the DG module controls the connection status of the DG unit. Moreover, a knowledge based managing of the network topologies is maintained by the broker module, which supports for matching the relevant DG modules in accordance with the requests from the relay modules.

In addition, the performance of the proposed scheme for coordinating the operations between relays and DG units is evaluated. Two simulation scenarios, including a multiple faults issue and a network topology change issue, are undertaken. The simulation results suggest the most valuable point of the proposed scheme is that it provides a flexible and scalable approach for anti-islanding protection of DG. Furthermore, a series of experiments is carried out for the investigation of the time consumption of agent communications in the proposed scheme. A multi-brokering mechanism is developed for reducing the response time of the broker module in handling a large number of requests. From the experiment results, the response time of the proposed scheme in tripping the DG units from the islanded condition is acceptable if there is not a large number relays being tripped at the same time. Further research will focus on improving the speed and reliability of agent communications in a distribution system, as well as developing more efficient agent-device interactions.

Chapter 5

Agent-based Information Management of a Substation

5.1 Introduction

This chapter introduces an agent-based substation information management system which is developed based on the proposed multi-agent framework. A few common issues, such as accessing to substation information, operation and maintenance strategies and substation asset management, are introduced first. The development of the proposed system is then described, including a system architecture, a number of agent-based modules and the tasks and protocols adopted by this system. Furthermore, an implementation of this system for substation asset management is presented. A substation network model is designed, followed by three specific applications, *i.e.,* relay and circuit breaker status monitoring, transformer condition assessment and data and document query.

5.1.1 Access to substation information

The application of intranet technology allows information related to the condition and performance of the primary plant, such as protection relays, circuit breakers, power transformers and other IEDs, to be accessed from the closed substation information systems. An information management unit, acting as a data base converter, plays a significant role in a substation which compresses data with closed format from substation equipment, converts data to information and provides this information via the LANs or WAN. Particularly, five issues are considered for accessing to substation information, which are summarised as follows:

- Process level for power generation, transmission, distribution and consumption
- Automation levels for power plant, transmission, distribution, and demand side automation, which comprise local control, object protection and on-line condition monitoring.
- Data exchange level for real time and non-real time transmission of data for control, fault and condition assessment, performance statistics, *etc.*, from remote.
- Support level to provide people who are responsible for system planning, operation, maintenance and asset management with direct access to disturbance records, event and alarm lists as well as condition related data of primary equipment.
- Management level to support energy management, transmission, and distribution as well as demand side management with accurate and fast real time data to allow immediate response to instabilities in order to maintain power system integrity.

5.1.2 Substation maintenance strategies

Maintenance is one of the existing tools to ensure satisfactory component and system reliability. Others include increasing system capacity, reinforcing redundancy and employing more reliable components. Furthermore, it is expected that effective maintenance policies can reduce the frequency of service interruptions and the undesirable consequences of such interruptions. Here, two particular issues are considered, which are data acquisition and transformer condition.

Data acquisition

With computing power making its way into the primary equipment, more and more equipment internal data can be made available to the outside at virtually no cost. Data that can be accessible includes switching counters, thermal information, quality of isolation media, entire timing curves of switching operations, manufacturing data, original value of key performance criteria *etc.*

Those data can be the source of valuable condition information and be exploited for building condition monitoring systems for those assets that exhibit the highest failure rates and/or cause unacceptable power interruption impact. Without doubt the transformers and circuit breakers are the prime candidates for these types of monitoring systems.

The second trend within the data acquisition falls into the category of IEDs, *i.e.* secondary equipment like protection terminals. Besides their primary functions, they host more and more additional functionalities, which increase their attractiveness compared with dedicated single function units. Many of these additional functions provide a sound foundation for basic monitoring systems, cost-efficient and are perfectly suited for medium and distribution voltage level IEDs for protection or control, such as disturbance or event recorders, statistical value recording, power quality analysers and general purpose programming capabilities that allow to write and run customer specific applications on the IEDs.

Transformer condition monitoring

Transformer unavailability has a considerable impact on the operation of electricity generation and installation networks. Monitoring systems available on the market already provide solutions for monitoring the conditions of windings and magnetic circuit (*e.g.* gas dissolved in oil, ultrasound emission, temperatures). They all aim to prevent major failures and extend service lifetime of the equipment by triggering preventive maintenance. Furthermore, the use of monitoring data must be coupled with the implementation of diagnostic tools.

In addition, the World Wide Web acts as an information source and its commercial application has created a mass market where the technology costs are shared by millions of developers and companies throughout the world. The costs for applying these mainstream technologies to other applications like transformer monitoring are relatively low when compared with proprietary solutions for networks, protocols, processors, and software environments.

5.1.3 Substation asset management

Asset management means operating a group of assets over the whole technical life-cycle guaranteeing a suitable return and ensuring defined service and security standards [115]. Distribution and transmission network operators are facing many different and partly even competing targets. For example, according to the pressure from both massive industrial growth and increasing capital expenditure, power system has been placed under unprecedented strains and face dilemmas with conflicting objectives on asset utilisation.

Asset management in electrical grid companies plays a key role in the detection and evaluation of decisions leading to a long-term economical success and best possible earnings. However a few challenges should be met in order to reach these expectations, as fundamental asset management covers aspects ranging from technical issues like network planning or the definition of operational fundamentals to more economical themes such as planning of investments and budgeting, and strategic planning issues.

5.2 Agent-based Substation Information Management System

5.2.1 System architecture

Based on the proposed multi-agent framework introduced in Chapter 2, an agent-based substation information management system is developed. Figure 5.1 illustrates the architecture of this system consisting of three parts, *i.e.*, a substation part, a control centre part and a distribution network part. Four modules are developed within this system, *i.e.*, a transformer module, a device module, an information aggregation module and a user interaction module. Each module is composed of a number of agents derived based on the generic

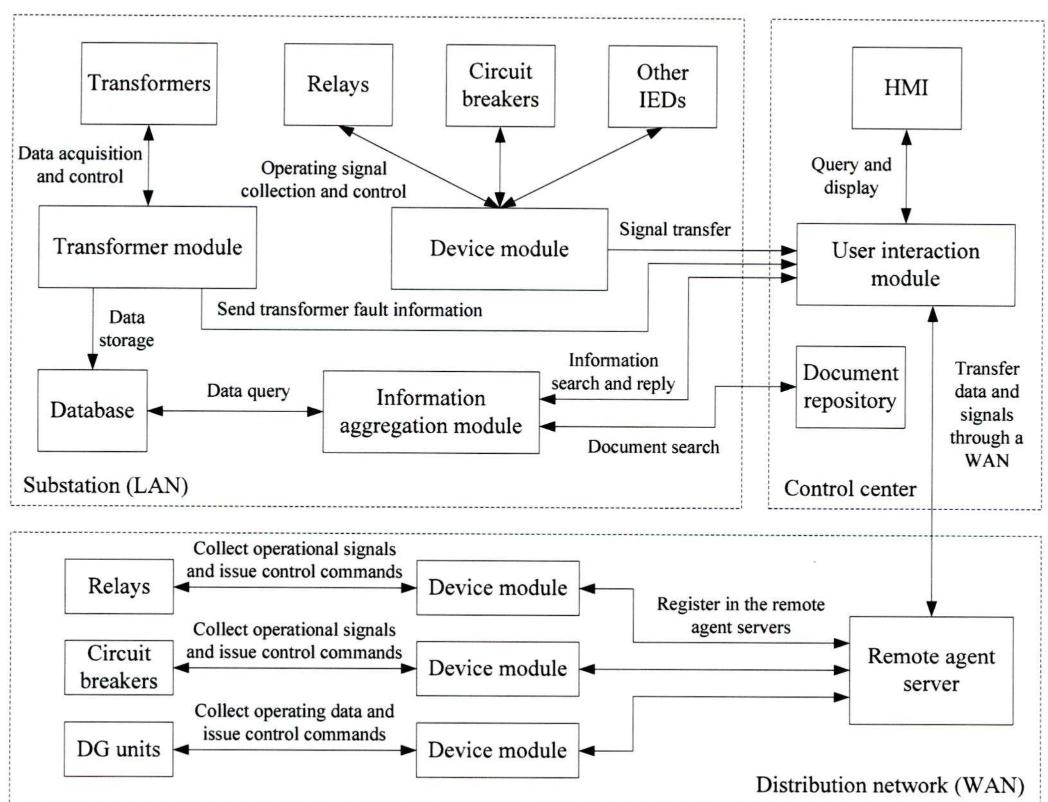


Figure 5.1: An architecture of agent-based substation information management system.

agent structure introduced in Section 2.3. In particular, the transformer modules, device modules and information aggregation modules are employed in the substation, the user interaction module is located in the control centre and a number of device modules are utilised in the distribution network. Communications between the substation and control centre are based on the substation LANs, while the modules employed in the distribution network must contact the remote agent server which is able to communicate with the control centre via WAN.

The functions and capabilities provided by the modules employed in the proposed system are listed as follows:

- Transformer module: it is able to acquire operational data through sensors installed in a power transformer in real-time and assess the working condition of the transformer using the specific fault diagnosis methods, *e.g.*, dissolved gas analysis (DGA), thermal methods *etc.*
- Device module: the device module is connected to the power devices, such as protection relays, circuit breakers and other IEDs, for collecting the operating signals and issuing control commands in accordance with the preset control principles in runtime.
- Information aggregation module: this module is capable of querying historical data from the databases and information management unit (IMU) and searching relevant documents from a document repository.
- User interaction module: an HMI is provided by this module for taking requests from the system operators and generating easy-to-understand dynamic diagrams and reports by use of the received device condition data.

5.2.2 Transformer module for fault diagnosis

The transformer module, defined as $\mathcal{M}_{\text{trans}}$, consists of four generic agents, *i.e.*, a control agent ($\mathcal{A}_{\text{CA\_T}}$), an analysis agent ($\mathcal{A}_{\text{AA\_T}}$), a communication agent

($\mathcal{A}_{\text{ComA\_T}}$) and a database agent ($\mathcal{A}_{\text{DBA\_T}}$). $\mathcal{M}_{\text{trans}}$ is expressed as:

$$\mathcal{M}_{\text{trans}} \supset \{\mathcal{A}_{\text{CA\_T}}, \mathcal{A}_{\text{AA\_T}}, \mathcal{A}_{\text{ComA\_T}}, \mathcal{A}_{\text{DBA\_T}}\}$$

The structure of $\mathcal{M}_{\text{trans}}$ is shown in Figure 5.2. Three functions are provided by this module, *i.e.*, receiving data from sensors installed in a power transformer in real-time and recording the data in a database, using association rule mining (ARM)-based DGA method for data analysis and condition assessment, and informing the user interaction module of the real-time working condition of the transformer.

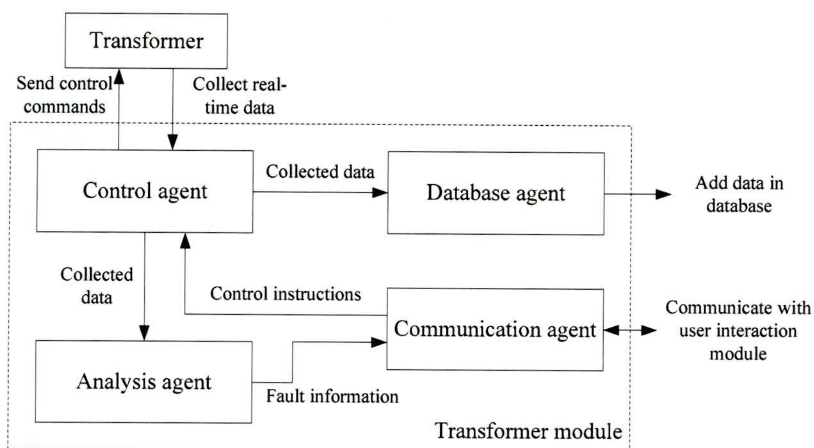


Figure 5.2: Structure of transformer module in agent-based substation information management system.

Association rule mining-based DGA method for transformer fault diagnosis

There are a number of specific methodologies and algorithms utilised by the software agents employed in the different modules for handling the issues in the proposed system. Particularly, an association rule mining (ARM)-based DGA method is utilised by the transformer module for transforming real-time condition assessment. DGA, as a standard on-line tool, has been used by the engineers to determine the condition of the power transformers. The main limitation of the standard methods for measuring the gas concentrations or ratios is that the pre-defined criteria are established based on the empirical studies, which may not fit within the predefined criteria, so that a fault diagnosis task may not be achievable. In order to solve these problems, an alternative method, ARM-based DGA method was proposed. The basic idea of implementing ARM in DGA is to generate association relationships between a set of key gas values and transformer working states, *i.e.*, various fault classes or no fault, from real fault diagnosis records. The derived association relationships are then interpreted as association rules. With these rules, the working state of transformers can be diagnosed.

Within this approach, each DGA record is composed of several key gas concentrations and a working state, which is derived by on-site inspections and converted into a new format, which employs the gas ratios used in a conventional DGA method to reorganise the key gas concentrations of the record. Furthermore, two methods, the Dornenburg ratio method and the Rogers ratio method, are utilised for generating the different gas ratios, respectively.

In addition, one item extracted from the DGA records is defined as an attribute. The main task of an ARM process is to discover the potential relationships and correlations, which are interpreted as a set of association rules, among user-interested attributes. In order to select the useful attributes, the gas ratios employed in a conventional DGA method have been chosen, because these gases are the chemical reaction products when the faults occur. As introduced previously, the available transformer operating states from the on-site inspections include thermal, PD, arcing and no fault.

Briefly, ARM can be described as follows: let I=$[i_1, i_2, \ldots, i_n]$ be a set of selected items, known as attributes. Define T to be a set of training data, *e.g.* DGA records. Each record R in T is composed of several items and assigned with a unique identifier. Also, $R \subseteq I$. Let A and B be two sets of items. A specific record R_t from T contains A if and only if $A \subseteq R_t$. An association rule is defined in the following form $A \rightarrow B$, where $A \subset I$, $B \subset I$ and $A \cap B = \varnothing$. In particular, A and B are defined as the antecedent and consequent of the rule respectively. The expression means that if A is presented in R_t, then B is likely presented in R_t as well.

With an ARM algorithm, two parameters, a support value and a confidence value, are specified for the generation of an association rule, which represent the percentage or amount of records in T that contain $A \cup B$ and the percentage of records in T holding A, that also contain B, respectively. Both of the two parameters can be expressed as: Support($A \rightarrow B$)=P($A \cup B$) and Confidence($A \rightarrow B$)=P($B|A$).

In addition, the rule set postprocessing methods, involving a rule set simplification method and a rule fitness evaluation method [116], are applied in ARM to select the useful rules in a fault classification task and assign a fitness value to each useful rule, respectively. A fitness value represents the reliability of a rule on a fault diagnosis task. In the useful rule set, a rule with a higher fitness value means that this rule is more reliable for a fault diagnosis task, compared with a rule holding a lower fitness value. Finally, with the useful rule set, an association rule-based fault diagnosis classifier, which is used for fault diagnosis of power transformers, is developed.

Real-time data acquisition and record

$\mathcal{A}_{\text{CA\_T}}$ employed in $\mathcal{M}_{\text{trans}}$ is to acquire the transformer operational data in real-time. The knowledge maintained $\mathcal{A}_{\text{CA\_T}}$ is presented as follows:

$$\mathcal{A}_{\text{CA\_T}}(\mathit{ID}, \mathit{Data\_type}, \mathit{Raw\_data})$$

where *ID* is the identification number of $\mathcal{A}_{\text{CA\_T}}$. The data types acquired from the sensors are defined as *Data_type* and a set of raw data received in each sampling is presented as *Raw_data*. In particular, *Data_type* is determined by the transformer condition assessment algorithms utilised by $\mathcal{A}_{\text{AA\_T}}$. *Raw_data* received in each time is added into the database by $\mathcal{A}_{\text{DBA\_T}}$ for recording the transformer functioning parameters and historical data queries.

Data processing and fault diagnosis

$\mathcal{A}_{\mathrm{AA\_T}}$ is capable of data analysis and condition assessment using the ARM-based DGA method introduced in Section 5.2.1. Generally, $\mathcal{A}_{\mathrm{AA\_T}}$ is presented as:

$$\mathcal{A}_{\mathrm{AA\_T}}(\mathit{ID}, \mathit{ARB}, \mathit{Data\_type}, \mathit{Dataset}, \mathit{Selected\_rule}, \mathit{Condition})$$

where *ID* is the identification number and *ARB* is the association rule base established based on the ARM-based DGA method. *Data_type* is defined in accordance with the requirement of *ARB* and forwarded to $\mathcal{A}_{\mathrm{CA\_T}}$ for filtering the real-data received from the sensors. *Dataset* is a set of processed raw data and a rule selected from the association rule based is defined as *Selected_rule*. Moreover, the determined real-time transformer condition is presented by the parameter *Condition*. Four steps are performed by $\mathcal{A}_{\mathrm{AA\_T}}$ for assessing the condition of power transformers, which are listed as follows.

Step 1: Data request Five types of transformer gas concentration data are requested by the utilised ARM-based DGA method, *i.e.*, Hydrogen (H_2), Methane (CH_4), Ethene (C_2H_6), Ethylene (C_2H_4), and Acetylene (C_2H_2). In this case, the parameter *Data_type* is defined as *Data_type* = {H2, CH4, C2H2, C2H4, C2H6}. Moreover, it is revisable if different condition assessment methods are used.

Step 2: Data processing Using the ARM-based DGA method, two methods, the Dornenburg ratio method and Rogers ratio method, are utilised for representing the association rules managed in the rule base. In this case, the received *Raw_data* must be calculated to meet the requirements of the utilised rule base. Specifically, *Datasets* processed by the two methods are presented as follows:

$$\begin{aligned} \mathit{Dataset}_{Dor} &= (\mathrm{CH4/H2, C2H2/C2H4, C2H2/CH4, C2H6/C2H2}) \\ \mathit{Dataset}_{Rog} &= (\mathrm{CH4/H2, C2H2/C2H4, C2H4/C2H6}) \end{aligned}$$

Step 3: Rule selection Using the processed *Dataset*, an optimal association rule can be selected from the rule base and represented by *Selected_rule*.

As introduced in Section 5.2.2, the ARM-based DGA method defines two parameters in each rule, *i.e.*, an antecedent and a fitness value. The rule is selected following the two principles:

1. The association rules whose antecedents are related to *Dataset* are given the priority;
2. The one rule with the highest fitness value is selected and presented as *Selected_rule*.

Particularly, in order to determine whether the antecedents of the association rules are related *Dataset*, R_{te} is defined as a vector $(A_1 \dots A_k \dots A_K)$, where A_k $(k = 1 \dots K)$ is the k_{th} attribute of R_{te}. A literal p is an attribute-value pair with the form $(A_k,\ v)$, where v is a possible value of the attribute A_k. Define that R_{te} satisfies a literal p if and only if $R_k = v$, where R_k is the value of A_k in R_{te}.

An association rule r is given as: $r = p_1 \wedge p_2 \wedge \dots \wedge p_l \rightarrow c$. In r, the antecedent is a conjunction of literals p_1, p_2,..., p_l and the consequent is the class c. The record R_{te} satisfies the antecedent of r if and only if it satisfies every literal of the antecedent of r. In the case that R_{te} satisfies r's antecedent, then the working state of R_{te} is classified as the class c. If a rule contains zero literal, its body is satisfied by any record.

For example, if *Dataset* $= (0.33, 0.81, 0.54, 0.11)$, the selected rule is presented as *Selected_rule* = "**If** $1.0 > CH_4/H_2 > 0.1$; **and** $C_2H_2/C_2H_4 > 0.75$; **and** $C_2H_2/CH_4 > 0.3$; **and** $C_2H_6/C_2H_2 < 0.4$, **Then** Arcing; **Fitness value**: 0.3592499".

Step 4: Condition assessment In the selected rule, the transformer working condition is described and can be presented by the parameter *Condition*. Generally, four condition types including "Normal", "Arcing", "PD" and "Thermal" are defined. In the above example, the current transformer condition is represented as *Condition* = "Arcing".

Transfer transformer condition information

The determined transformer condition *Condition* is forwarded to $\mathcal{A}_{\text{ComA\_T}}$ for transferring to the user interaction module. An ACL message is generated and sent to user interaction module by $\mathcal{A}_{\text{ComA\_T}}$. The contents of the message include "*ID, Condition*" and the performative is INFORM.

5.2.3 Device module for data acquisition and control

The device module, defined as $\mathcal{M}_{\text{deivce}}$, is connected with power devices, such as protection relays, circuit breakers, IEDs, *etc.*, for collecting runtime operating signals and performing control actions. As shown in Figure 5.3, $\mathcal{M}_{\text{deivce}}$ consists of three agents developed based on the generic structure introduced in Chapter 2, which can be presented as follows:

$$\mathcal{M}_{\text{device}} \supset \{\mathcal{A}_{\text{CA\_D}}, \mathcal{A}_{\text{ComA\_D}}, \mathcal{A}_{\text{DBA\_D}}\}$$

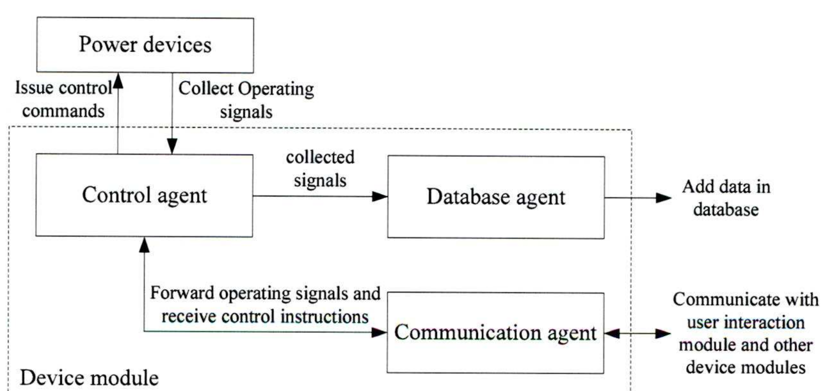


Figure 5.3: Structure of device module in agent-based substation information management system.

$\mathcal{M}_{\text{deivce}}$ is able to receive the operational signals from the connected power device and manage these signals in the databases. It is also capable of communicating with the user interaction module or the other device modules as well as issuing control commands to the power device for coordinating its operations. In particular, the relay module and the DG module for the protection of distribution networks introduced in Chapters 3 and 4 are both derived from the device module.

Collect operating signals

$\mathcal{A}_{\text{CA\_D}}$ is able to receive operating signals from the connected power devices, such as relays and breakers. A parameter *Device_status* is defined to indicate the runtime condition of the connected device. Particularly, if the connected device is a protection relay or a circuit breaker, *Device_status* is represented as:

$$Device\_status = \begin{cases} \text{OPEN} & \text{if a relay is tripped or a breaker is open} \\ \text{CLOSE} & \text{if a relay or a breaker is reclosed} \end{cases}$$

Once the device is operated, *Device_status* is changed and forwarded to $\mathcal{A}_{\text{ComA\_D}}$ for communicating with the user interaction module or other device modules. Furthermore, the received signals are recorded in the database by $\mathcal{A}_{\text{DBA\_D}}$, which can be queried and extracted by the information aggregation module.

Communicate with other modules

The collected operating signals need to be transferred to the user interaction module and other device modules employed in the proposed system for informing the runtime status of the monitored power device and coordinating their operations, respectively. An ACL message is generated by $\mathcal{A}_{\text{ComA\_D}}$ with contents of "*ID, Device_status*" and the performative is INFORM. Moreover, the reply messages are received by $\mathcal{A}_{\text{ComA\_D}}$ from both user interaction and device modules.

Issue control commands

$\mathcal{A}_{\text{CA\_D}}$ is capable of issuing control commands to the connected power device for coordinating its actions in accordance with the pre-defined control principles. The detailed description of this function provided by $\mathcal{M}_{\text{device}}$ are given in Sections 3.2 and 4.2.

5.2.4 Information aggregation module for data query and document search

The information aggregation module developed in the proposed system is defined as $\mathcal{M}_{\text{info}}$, which plays an important role in data query and document retrieval. $\mathcal{M}_{\text{info}}$ can be presented as follows:

$$\mathcal{M}_{\text{info}} \supset \{\mathcal{A}_{\text{ComA\_I}}, \mathcal{A}_{\text{DocA\_I}}, \mathcal{A}_{\text{DBA\_I}}\}$$

Three generic agents are employed in this module, including a communication agent ($\mathcal{A}_{\text{ComA\_I}}$) for receiving requests from the user interaction module and replying the results, a database agent ($\mathcal{A}_{\text{DBA\_I}}$) for querying the connected database and a document agent ($\mathcal{A}_{\text{DocA\_I}}$) for searching the relevant documents in the document repository in accordance with the received requests. Figure 5.4 illustrates the structure of this module.

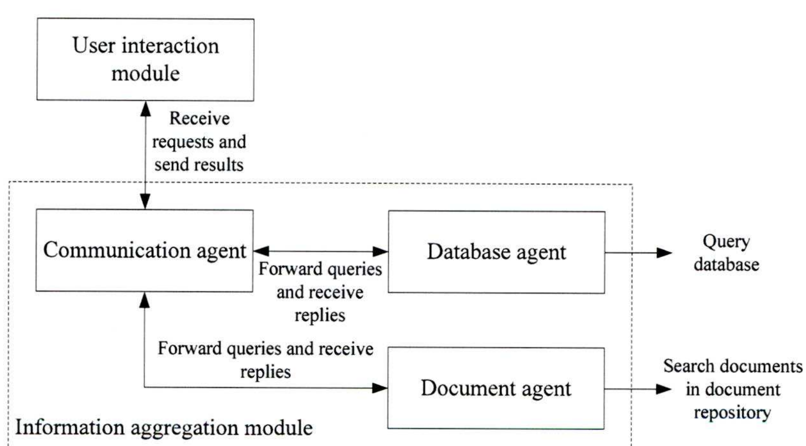


Figure 5.4: Structure of information aggregation module in agent-based substation information management system.

Communicate with user interaction module

$\mathcal{A}_{\text{ComA\_I}}$ is able to communicate with the user interaction module for receiving the requests and replying the results. Specifically, an ACL request message is received by $\mathcal{A}_{\text{ComA\_I}}$, including "*ID_ComA_U, Request, Type*", where

ID_ComA_U is the identification number of the communication agent employed in the user interaction module. The contents and type of a request are presented as *Request* and *Type*, respectively. In particular, two types of requests are carried out by $\mathcal{A}_{\text{ComA\_I}}$, including data and document. In this case, if *Type* = "data" *Request* will be forwarded to $\mathcal{A}_{\text{DBA\_I}}$ for data query, while *Request* is transferred to $\mathcal{A}_{\text{DocA\_I}}$ for searching the relevant documents in the document repository, if *Type* = "document".

Data query and document retrieval

Data query process is carried out by $\mathcal{A}_{\text{DBA\_I}}$ which maps the received request into an SQL query which is then sent to the database. The results are converted into FIPA ACL / FIPA SL and saved as *Data_result* which will be returned to $\mathcal{A}_{\text{ComA\_I}}$.

In addition, document retrieval is performed by $\mathcal{A}_{\text{DocA\_I}}$ that maintains an ontology-based document search engine for searching most relevant documents in the document repository. Briefly, ontology is a knowledge representation mechanism for semantically depicting information with their meanings and relationships. Ontology descriptions are utilised by $\mathcal{A}_{\text{DBA\_I}}$ to annotate and expand the received requests. Furthermore, an evidential reasoning (ER) method, known as "Dempster-Shafer" theory is utilised for evidence combination of the developed Multiple Attribute Decision Making (MADM) tree model which is generated from the terms of the expanded *Request*. Accordingly, the results are saved as *Document_result* and returned to $\mathcal{A}_{\text{ComA\_I}}$.

5.2.5 User interaction module for user-system interaction

The user interaction module, defined as $\mathcal{M}_{\text{user}}$, provides a friendly HMI for system operator accessing information resources and observing user's preference optionally to satisfy the user's specific requirements. As illustrated in Figure 5.5, two generic agents are employed in this module, *i.e.*, an interface agent

($\mathcal{A}_{\text{IA\_U}}$) for taking information from the HMI and displaying the diagrams and reports generated from the received information and a communication agent ($\mathcal{A}_{\text{ComA\_U}}$) which is able to communicate with all the other modules employed in the proposed system.

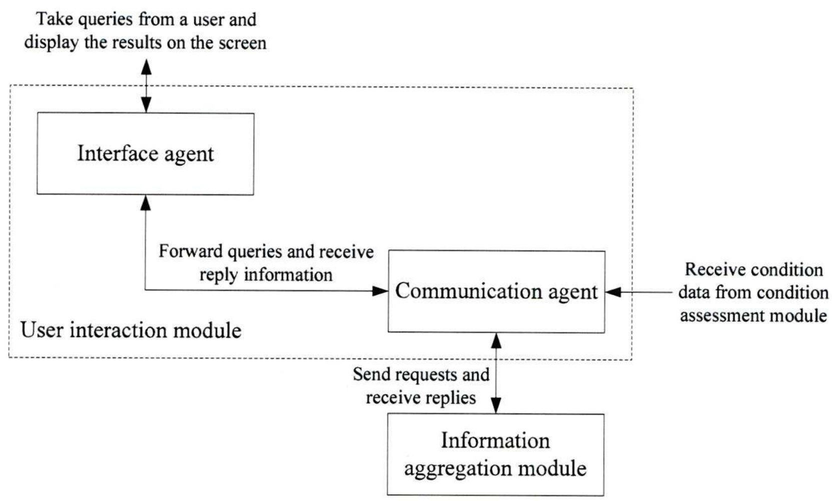


Figure 5.5: Structure of user interaction module in agent-based substation information management system.

User interface

User system interaction is carried out by $\mathcal{A}_{\text{IA\_U}}$, which provides a web-based HMI for taking commands and queries from the user and translating them into appropriate ACL messages for transmission to other modules. Moreover, the received raw data or signals are converted to easy-to-understand diagrams or reports by the visualisation tools integrated with $\mathcal{A}_{\text{IA\_U}}$ and displayed on screen.

Communicate with other modules

$\mathcal{A}_{\text{ComA\_U}}$ is able to send ACL messages to all the other modules and receive replies. For example, it is capable of requesting the information aggregation module for data query and document retrieval, receiving device operating status signals from and issuing control instructions to the device modules, as well as receiving real-time transformer working condition parameters from the transformer module. In order to distinguish the messages received from the different modules, a number of ACL message templates are defined by $\mathcal{A}_{\text{ComA\_U}}$ using the performative and conversation ID to categories them.

Communicate with transformer module: ACL messages can be received from the transformer module if they match Template 1. The performative of this message should match INFORM and the conversation ID should be "ID_ComA_T".

Template 1 ACL message template received from $\mathcal{A}_{\text{ComA\_T}}$

```
(MessageTemplate mt_transformer =
        MessageTemplate.and(
                MessageTemplate.MatchPerformative(ACLMessage.INFORM),
                MessageTemplate.MatchConversationID(''ID_ComA_T'')))
```

Communicate with device module: The performative and conversation ID of the messages received from $\mathcal{A}_{\text{ComA\_D}}$ must be INFORM and "ID_ComA_D", respectively, which should match Template 2.

Template 2 ACL message template received from $\mathcal{A}_{\text{ComA\_D}}$

```
(MessageTemplate mt_device =
        MessageTemplate.and(
                MessageTemplate.MatchPerformative(ACLMessage.INFORM),
                MessageTemplate.MatchConversationID(''ID_ComA_D'')))
```

Communicate with information aggregation module: Template 3 is utilised for matching the messages received from the information aggregation module.

Template 3 ACL message template received from $\mathcal{A}_{\mathrm{ComA\_I}}$

```
(MessageTemplate mt_information =
          MessageTemplate.and(
                   MessageTemplate.MatchPerformative(ACLMessage.INFORM),
                   MessageTemplate.MatchConversationID(``ID_ComA_I''')))
```

5.2.6 Tasks and interaction protocols

Each task described in this section is associated with a particularly type of data and a particular interaction protocol, taken from the standard FIPA interaction protocols. Tables 5.1 and 5.2 summarise these associations. In particular, Table 5.1 considers each task as a transformation from input data to output data, and defines the types of data involved and the agents that perform the task; Table 5.2 considers the characteristics of the task (whether it is event-driven or performed on demand by user or system) and the protocols involved.

5.3 Implementation of Proposed System in Substation Asset Management

This section provides several examples of how the proposed system is implemented in substation asset management. Various tasks are performed, including monitoring relay and breaker operations in protection status, transformer condition assessment, data query and document retrieval. Messages transferred among the agents are provided to demonstrate the use of the FIPA standard protocols. In particular, each ACL message has a sender, receiver, content (which is written in the FIPA SL) and protocol. In these examples, the agent name `ID@chenma:1099/JADE` represents the name of an agent and `conversation-id` allows the participants to distinguish between several concurrent conversations. Certain message parameters (*e.g.*, encoding, ontology) have been omitted from the messages. Each time an agent wishes to carry out

Table 5.1: Tasks, data types and agents.

Task	Input Data	Output Data	Agents
User interaction			
- Queries	HMI input	ACL queries	$\mathcal{A}_{\text{ComA\_U}}$
- Requests	HMI input	ACL queries	$\mathcal{A}_{\text{ComA\_U}}$
- Online display	Events or status	HMI display	$\mathcal{A}_{\text{IA\_U}}$
Data querying	Queries (ACL)	Responses (ACL)	$\mathcal{A}_{\text{DBA\_I}}$
Data Storage	Data in ACL format and global ontology	Data in SQL statements and database schema	$\mathcal{A}_{\text{DBA\_T}}$ $\mathcal{A}_{\text{DBA\_D}}$
Document retrieval	Queries (ACL)	Binary data	$\mathcal{A}_{\text{DocA\_I}}$
Document storage	Binary data in ACL	Binary data in files	$\mathcal{A}_{\text{DocA\_I}}$
Data Acquisition	Sensor data	Events/status	$\mathcal{A}_{\text{CA\_T}}$ $\mathcal{A}_{\text{CA\_D}}$
Output D.I.	Required device status	DAQ actions	$\mathcal{A}_{\text{CA\_T}}$ $\mathcal{A}_{\text{CA\_D}}$
Input D.I.	DAQ Events/ Status	Device Events/ Status	$\mathcal{A}_{\text{CA\_T}}$ $\mathcal{A}_{\text{CA\_D}}$
Analogue/Digital Output	New values (ACL)	DAQ Channel Values	$\mathcal{A}_{\text{CA\_T}}$ $\mathcal{A}_{\text{CA\_D}}$

an action or to submit a query, it will first search the DF to find other agents capable of processing that request or answering that query, unless an appropriate agent is already known and is still available (has not been disconnected). This means that it is possible to substitute different agents providing the same information, and permits agents to be added and removed at runtime.

Table 5.2: Tasks and interaction protocols.

Task	Characteristics	Protocols
User Interaction		
-Queries	On-demand	FIPA Query
-Requests	Request/Reply	FIPA Request
-Online Display	Event-driven	FIPA Subscribe
Intervention	Request/Reply	FIPA Request
Output D.I.	On request	FIPA Request
Input D.I.	Event-driven	FIPA Subscribe
Data Acquisition	Event-driven	Device-dependent(input) FIPA Subscribe (output)
Data Storage	Event-driven	FIPA Subscribe (data gathering)
Data Querying	On-Demand	FIPA Query
Document Retrieval	On-Demand	FIPA Request,FIPA Query
Document Storage	On-Demand	FIPA Request
Information Gathering	On-Demand	FIPA Query, (FIPA Request)
Output (A or D)	On-Demand	FIPA Request

5.3.1 Substation network module

It is impossible to test all scenarios as real data from the control centre and IEDs are difficult to obtain. To partially solve this problem, a typical 132/11 kV substation given in Figure 5.6 and its associated distribution network has been simulated in PSCAD/EMTDC programme environment. This model consists of the primary plant and the secondary protection and control systems. Two transformers are protected by the directional overcurrent relays (R_3 and R_4) which prevent both transformers tripping when a fault occurs on the 11 kV terminals. Four non-directional time graded overcurrent relays (R_5, R_6, R_7 and R_8) are installed on the outgoing feeders, where two DG units (DG_1 and DG_2) are integrated. Furthermore, the 132/11 kV transformers are protected by restricted earth fault (REF) and biased differential unit protection (BDF). A selection of fault scenarios are applied to the simulator.

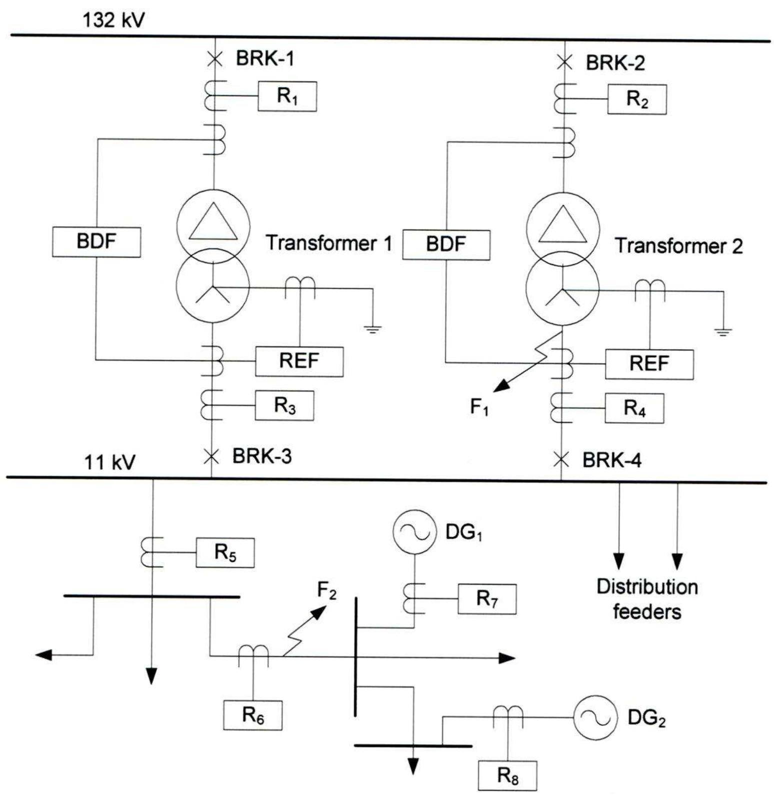


Figure 5.6: A 132/11 kV substation and distribution network model.

The proposed system is implemented on this network model. Specifically, the user interaction module is executed in the agent server located in the control centre. The other three modules, including the transformer modules, device modules and information aggregation modules are employed in the substation communicating with the user interaction module via the LAN. Moreover, a number of device modules are connected with the relays, circuit breakers and DG units in the distribution network and registers in the remote agent servers for communicating with the user interaction module through the WAN.

In particular, the operating response of the relays circuite breakers, the real-time gas concentration data acquired from the two transformers and voltage and current collected from other sensors are recorded and stored in a database. Based on this model, three applications of the proposed system are discussed, namely, management of status of protection relays and circuit breakers, transformer condition assessment and information gathering. They are introduced in the following three subsections in detail.

5.3.2 Monitoring relay operations

Operational signals of a protection relay or a circuit breaker are collected by the device module. As introduced in Section 5.2.3, the parameter *Device_status* defines the status of the relay or breaker. In this example, for a circuit breaker, the value set is either open (1) or close (0), while relay operations are defined as "pickup" and "trip". Two scenarios are considered in this example, including a directional A - B fault (F1) and a three phase fault (F2), which are shown in Figure 5.6 respectively.

Directional fault on F1

The first case considers the effect of a solid directional A - B fault which occurs at F1 at 1 s. According to the characteristics of R_4, it is tripped at 1.119 s and BRK-4 and BRK-2 are opened at 1.133 s and 1.139 s accordingly. The operating signals can be collected by $\mathcal{A}_{\text{CA\_R4}}$, $\mathcal{A}_{\text{CA\_B4}}$ and $\mathcal{A}_{\text{CA\_B2}}$ and saved in the database. Table 5.3 illustrates the detailed operating times of R_4, BRK-4 and BRK-2.

Table 5.3: Protection status when a fault occurs at F_1.

Time	R4		BRK-4	BRK-2
t(s)	Pickup	Trip	Open	Open
1.039 s	1	0	0	0
1.119 s	1	1	0	0
1.133 s	1	1	1	0
1.139 s	1	1	1	1

Three phase fault on F2

The second scenario considers a three phase fault at 1.0 s on F2 located in the distribution network. Following the protection settings, R_6 is tripped at 1.115 s and $\mathcal{A}_{\mathrm{CA\_R6}}$ receives the trip signal at the same time. As introduced in Chapter 4, DG_1 and DG_2 are islanded after 1.115 s due to the loss of mains. Accordingly, the device modules connected to both R_7 and R_8 communicate with the device module connected to R_6. In this case, $\mathcal{A}_{\mathrm{ComA\_R6}}$ sends the received trip signal to both $\mathcal{A}_{\mathrm{ComA\_R7}}$ and $\mathcal{A}_{\mathrm{ComA\_R8}}$ with Message 1:

Message 1 An ACL message sent by $\mathcal{A}_{\mathrm{ComA\_R6}}$

`(Inform`	An information message
`  :sender`	
`    (agent-identifier`	Sent by $\mathcal{A}_{\mathrm{ComA\_R6}}$
`        :name ComA_R6@pc2214:1099/JADE`	
`  :receiver`	Receiver is $\mathcal{A}_{\mathrm{ComA\_R7}}$
`    (set (agent-identifier To`	and $\mathcal{A}_{\mathrm{ComA\_R8}}$
`          :name ComA_R7@pc2214:1099/JADE`	
`                ComA_R8@pc2214:1099/JADE))`	
`  :content`	Send relay signals
`    ''Relay_status = OPEN''`	
`  :protocol fipa-inform)`	FIPA Inform protocol

Following the received message, $\mathcal{A}_{\mathrm{CA\_R7}}$ and $\mathcal{A}_{\mathrm{CA\_R8}}$ disconnect the associated DG units immediately. Consequently, R_7 and R_8 are tripped both at 1.225 s. Table 5.4 demonstrates the protection status of R_6, R_7 and R_8 in detail. After 0.5 s, R_6 is set to be reclosed. The reclosure signal is then received by $\mathcal{A}_{\mathrm{CA\_R6}}$ and forwarded to $\mathcal{A}_{\mathrm{ComA\_R6}}$ which informs both $\mathcal{A}_{\mathrm{ComA\_R7}}$ and $\mathcal{A}_{\mathrm{ComA\_R8}}$ with Message 2

In accordance with the received reclosure signal, $\mathcal{A}_{\mathrm{ComA\_R7}}$ and $\mathcal{A}_{\mathrm{ComA\_R8}}$ send reclosure commands to the associated relays to reconnect DG_1 and DG_2. In this case, the overall system can be restored after 1.825 s.

Table 5.4: Protection status when a fault occurs at F_2.

Time	R6		R7		R8	
t(s)	Pickup	Trip	Pickup	Trip	Pickup	Trip
1.095 s	1	0	0	0	0	0
1.115 s	1	1	0	0	0	0
1.205 s	1	1	1	0	1	0
1.225 s	1	1	1	1	1	1
1.595 s	0	1	1	1	1	1
1.615 s	0	0	1	1	1	1
1.805 s	0	0	0	1	0	1
1.825 s	0	0	0	0	0	0

Message 2 An ACL message sent by $\mathcal{A}_{\text{ComA\_R6}}$

`(Inform`	An information message
`  :sender`	
`    (agent-identifier`	Sent by $\mathcal{A}_{\text{ComA\_R6}}$
`        :name ComA_R6@pc2214:1099/JADE`	
`  :receiver`	Receiver is $\mathcal{A}_{\text{ComA\_R7}}$
`    (set (agent-identifier To`	and $\mathcal{A}_{\text{ComA\_R8}}$
`            :name ComA_R7@pc2214:1099/JADE`	
`                  ComA_R8@pc2214:1099/JADE))`	
`  :content`	Send relay signals
`    ''Relay_status = CLOSE''`	
`  :protocol fipa-inform)`	FIPA Inform protocol

5.3.3 Transformer condition assessment

As introduced in Section 5.2.2, transformer condition assessment is performed by the transformer module. In this application, two association rule bases (known as ARB-1 and ARB-2) are established based on the Dornenburg ratio method and Rogers ratio method, respectively. Totally 1016 DGA records are utilised to generate the association rules. In order to obtain the highest diagnosis accuracy in the proposed system, the association rules are generated with the various sets of the support and the confidence values, and a set of the most accurate rules are selected as the rule base.

When ARB-1 is utilised by $\mathcal{A}_{\text{AA\_T}}$, the acquired data are calculated based on the Dornenburg ratio method and saved as *Dataset1*, which are presented as follows:

$$\begin{aligned} Dataset1 &= (\mathrm{CH_4/H_2, C_2H_2/C_2H_4, C_2H_2/CH_4, C_2H_6/C_2H_2}) \\ &= (0.33, 0.81, 0.54, 0.11) \end{aligned}$$

Using *Dataset1*, two association rules are selected from ARB-1 by $\mathcal{A}_{\text{AA\_T}}$, which are listed as follows:

Rule 2: If $1.0 > \mathrm{CH_4/H_2} > 0.1$; and $\mathrm{C_2H_2/C_2H_4} > 0.75$; and $\mathrm{C_2H_2/CH_4} > 0.3$; and $\mathrm{C_2H_6/C_2H_2} < 0.4$, **Then** Arcing. **Fitness value**: 0.3592499.

Rule 5: If $1.0 > \mathrm{CH_4/H_2} > 0.1$; and $\mathrm{C_2H_2/C_2H_4} > 0.75$; and $\mathrm{C_2H_2/CH_4} > 0.3$; and $\mathrm{C_2H_6/C_2H_2} < 0.4$, **Then** PD. **Fitness value**: 0.1151944.

Comparing the fitness values of the two association rules, **Rule 2** with the higher fitness value is chosen from the rule base and defined as *Selected_rule1* The fault information is extracted from this rule and presented as *Condition1* = "Arcing", which is forwarded to $\mathcal{A}_{\text{DBA\_T}}$ for saving this event in the database and transferred to the user interaction module for alarming the operators that the transformer is under fault condition. In particular, an information message is sent by $\mathcal{A}_{\text{ComA\_T}}$ to the user interaction module using Message 3

Adding a new association rule base

Another example is performed to demonstrate the flexibility and autonomy of the proposed system. A new association rule base (ARB-2) established based on the Rogers ratio method is intended to be integrated with the existing system in runtime and the current used association rule base (ARB-1) will be deregistered. In this case, the data types requested by $\mathcal{A}_{\text{AA\_T}}$ will be changed.

Message 3 An ACL message sent by $\mathcal{A}_{\mathrm{ComA\_T}}$

`(Inform`	An information message
`  :sender`	
`    (agent-identifier`	Sent by $\mathcal{A}_{\mathrm{ComA\_T}}$
`        :name ComA_T1@pc2214:1099/JADE`	
`  :receiver`	
`    (set (agent-identifier To`	Receiver is $\mathcal{A}_{\mathrm{ComA\_U}}$
`            :name ComA_U@pc2214:1099/JADE))`	
`  :content ''Condition1 = Arcing''`	
`  :protocol fipa-inform)`	FIPA Inform protocol
`  :conversation-id ID_ComA_T)`	Set Conversation ID

Agents employed in the transformer module are capable of handling this issue and reconfiguring themselves automatically. According to the requirements of ARB-2, a new dataset is defined as

$$Dataset2 \;= (\mathrm{CH_4/H_2}, \mathrm{C_2H_2/C_2H_4}, \mathrm{C_2H_4/C_2H_6})$$

The received real-time gas concentration data will be recalculated by $\mathcal{A}_{\mathrm{AA\_T}}$. For example, a new dataset is processed as *Dataset2* = (0.72, 2.12, 4.76). Accordingly, a number of related association rules are matched in ARB-2. Following the rule selection process introduced in Section 5.2.2, the parameter *Selected_rule2* is defined by $\mathcal{A}_{\mathrm{AA\_T}}$ and presented as:

> ***Selected_rule2*** = "**If** $0.1 \leq \mathrm{CH_4/H_2} \leq 1.0$; and $\mathrm{C_2H_2/C_2H_4} < 0.1$; and $\mathrm{C_2H_4/C_2H_6} \leq 1.0$. **Then** PD. **Fitness value**: 0.4122782"

The current transformer condition is then defined as *Condition* = PD, which will be sent to the user interaction module for alerting the fault of the transformer with Message 4

Message 4 An ACL message sent by $\mathcal{A}_{\text{ComA\_T}}$

```
(Inform                                             An information message
  :sender
    (agent-identifier                               Sent by A_ComA_T
        :name ComA_T@pc2214:1099/JADE
  :receiver
    (set (agent-identifier To                       Receiver is A_ComA_U
            :name ComA_U@pc2214:1099/JADE))
  :content ''Condition2 = PD''
  :protocol fipa-inform)                            FIPA Inform protocol
  :conversation-id ID_ComA_T)                       Set Conversation ID
```

5.3.4 Data query and document retrieval

Query historical data

In order to assist a user in generating a query, the proposed system provides a series of steps for query generation. Once the user has selected the object, property, start time and end time of the data set in the HMI, this information is converted into a FIPA ACL query by $\mathcal{A}_{\text{IA\_U}}$ and forwarded to $\mathcal{A}_{\text{ComA\_U}}$. For example, suppose that the user has selected "relay1" as the object, "trip" as the property, 1/3/09 12:00:00 as the start time and 5/3/09 12:00:00 as the end time. $\mathcal{A}_{\text{ComA\_U}}$ will then query $\mathcal{A}_{\text{ComA\_I}}$ with Message 5

$\mathcal{A}_{\text{ComA\_I}}$ will then passed this message to $\mathcal{A}_{\text{DBA\_I}}$ that is to convert this into a SQL sentence using its reasoning engine for retrieving the relevant data from the database. The retrieved data will be passed back to $\mathcal{A}_{\text{ComA\_U}}$ with Message 6 and $\mathcal{A}_{\text{IA\_U}}$ then displays the data on the HMI as a graph or a table.

Search documents

To search for documents, the user first inputs a set of keywords into the HMI. These are then transmitted to $\mathcal{A}_{\text{IA\_U}}$ via the DataSocket connection. $\mathcal{A}_{\text{IA\_U}}$ then forward this requests to $\mathcal{A}_{\text{ComA\_U}}$ which sends a message to the

Message 5 An ACL message sent by $\mathcal{A}_{\text{ComA\_U}}$

(query-ref	Query message
:sender	
(agent-identifier	Sent by $\mathcal{A}_{\text{ComA\_U}}$
:name ComA_U@pc2214:1099/JADE	
:receiver	
(set (agent-identifier To	Receiver is $\mathcal{A}_{\text{ComA\_I}}$
:name ComA_I@pc2214:1099/JADE))	
:content ''((Type = Document) (all (set ?a ?t)	
(and (t (relay1 trip ?a) ?t)	Query relay trip signals
(sequence ?d ?r))))''	
:protocol fipa-query)	FIPA Query protocol
:conversation-id ID_ComA_U)	Set Conversation ID

Message 6 An ACL message sent by $\mathcal{A}_{\text{ComA\_I}}$

(inform	Information message
:sender	
(agent-identifier	Sent by $\mathcal{A}_{\text{ComA\_I}}$
:name ComA_I@pc2214:1099/JADE	
:receiver	
(set (agent-identifier to	Receiver is $\mathcal{A}_{\text{ComA\_U}}$
:name ComA_U@pc2214:1099/JADE))	
:content ''((result <action> (resource	Send data column
:property <relay trip signals>	
:format <format>	
:content <data>	
:protocol fipa-inform)	FIPA Query protocol
:conversation-id ID_ComA_I)	Set Conversation ID

information aggregation module, requesting it to inform $\mathcal{A}_{\text{IA\_U}}$ of all documents relevant to that query. For example, suppose that the query chosen by the user was "transformer monitoring". Message 7 is then sent by $\mathcal{A}_{\text{ComA\_U}}$.

Message 7 An ACL message sent by $\mathcal{A}_{\text{ComA\_U}}$

`(request`	Request message
`:sender`	
`(agent-identifier`	Sent by $\mathcal{A}_{\text{ComA\_U}}$
`:name ComA_U@pc2214:1099/JADE`	
`:receiver`	
`(set (agent-identifier To`	Receiver is $\mathcal{A}_{\text{ComA\_I}}$
`:name ComA_I@pc2214:1099/JADE))`	
`:content`	Send requests
`''((all (Type = Document)`	
`(sequence ?d ?r)`	
`(relevance ?d (set ''transformer''`	
`''monitoring'') ?r)))''`	
`:protocol fipa-request)`	FIPA Request protocol
`:conversation-id ID_ComA_U)`	Set Conversation ID

Using the proposed ontology-based search engine, the received requests are expanded and the relevant documents are retrieved from the document repository. A reply message following Message 8 is sent from $\mathcal{A}_{\text{ComA\_I}}$, giving the resource descriptions of any relevant documents. The names and relevancies of these documents are passed to $\mathcal{A}_{\text{ComA\_U}}$ and displayed in the HMI.

5.4 Summary

This chapter describes a substation information management system developed based on the proposed multi-agent framework. The development of the system architecture and the related technology, ARM-based DGA method, for

Message 8 An ACL message sent by $\mathcal{A}_{\mathrm{ComA\_I}}$

```
(inform                                                   Information message
  :sender
    (agent-identifier                                     Sent by A_ComA_I
        :name ComA_I@pc2214:1099/JADE
  :receiver
    (set (agent-identifier to                             Receiver is A_ComA_U
            :name ComA_U@pc2214:1099/JADE))
  :content                                                Send requests
    ``((all (Type = Document) (sequence ?d ?r)
              (relevance ?d (set ``transformer''
                     ``monitoring'') ?r)))''
              (set (sequence description
                       :title iSCSBrA4) 0.9705195)
                 (sequence (description
                       :title ``SS7 - Bricker'') 0.95818204))
  :protocol fipa-query)                                   FIPA Query protocol
  :conversation-id ID_ComA_I)                             Set Conversation ID
```

transformer fault diagnosis are introduced. Four modules, namely as a transformer module, a device module, an information aggregation module and a user interaction module, are designed under this architecture. Each module consists of a number of generic agents. Furthermore, the implementation of the proposed system in substation asset management is presented. Three specific applications are introduced, *i.e.* protection status monitoring, transformer condition assessment and information retrieval. In particular, a substation network module together with its distribution network is designed for simulation purpose. All of the messages transferred among the agents and the tasks and protocols carried out by the agents employed in this system are described in detail to demonstrate the feasibility of the system.

Chapter 6

Conclusion and Future Work

This chapter concludes the thesis and summarises the major achievements of the presented research work in the field of agent-based protection, fault diagnosis and asset management for power system automation. The following section provides a summary of all the results obtained and details of the major contributions. Then the challenges of utilising MAS for power system automation are discussed. Finally, the suggestions for future research are presented.

6.1 Summary

This thesis has described the development of a multi-agent framework for distribution network protection and substation information management. Within this framework, a number of software agents has been developed based on a generic structure for data acquisition, data processing, information gathering, control and communication. A reasoning engine and three unified units are integrated with the generic structure to support agent decision making and operations. All the agents are registered in the JADE platform following FIPA agent development standards. Based on this framework, an agent-based relaying scheme and an agent brokering-based anti-islanding protection scheme have been designed for the protection of distribution network with DG integrated, and an agent-based substation information management system has been developed and implemented in substation asset management. Specifically, in the preceding chapters, the following work and results were presented.

1. A background of power system automation, particularly for protection and information management and an introduction to agents and multi-agent systems, including the definitions, architectures, development standards and applications in power system automation were given in Chapter 1.

2. In Chapter 2 the development of a multi-agent framework for distribution network protection and substation information management was presented. The functions and specifications provided by the software agents employed this framework were then introduced. The generic structure that defines an agent kernel in which a reasoning engine is built for making decisions and three unified units, a knowledge unit, an input unit and an operation unit for supporting agent actions was described. Agent communication protocols utilised in this framework were also introduced. Moreover, a description of the developed agent brokering mechanism, including the structure of a broker agent and its communication patters, was given followed by the simulation study on the investigation of the timing performance of the broker agent.

3. Based on the proposed multi-agent framework, a novel agent-based relaying scheme was introduced in Chapter 3 and an agent brokering-based anti-islanding approach was described in Chapter 4, which provided the flexible and autonomous solutions for protecting a distribution network with DG integrated. Agent modules, such as a relay module and a DG module, were designed to connect with the protection relays and DG units for collecting real-time operational signals and supporting coordination among these devices. Moreover, a broker module was employed to improve communications among the agents in the agent brokering-based anti-islanding protection scheme. A number of simulation studies were carried for evaluating the performance of the proposed two schemes. Different protection issues, such as overcurrent protection and impedance

protection, and fault types, such as phase-to-ground faults and phase-to-phase faults were considered. From the simulation results, The merits of the presented agent-based schemes are in the flexibility, scalability and dynamic response for the protection of a distribution system. It also brings additional value in understanding how DG contribution to the system capacity, dynamics and operational requirements of a protection relay can be better controlled.

4. Chapter 5 presented an agent-based substation information management system which is developed based on the proposed multi-agent framework. The system architecture and the related technology utilised in this system were introduced. Four agent modules, such as a transformer module, a device module, an information aggregation module and a user interaction module, are developed within this system were also presented. Furthermore, the implementation of this system in substation asset management was introduced. Three specific applications were presented, such as protection status monitoring, transformer condition assessment and information retrieval. In particular, a substation network module was designed with its distribution network for simulation of these applications. All of the messages transferred among the agents and the tasks and protocols carried out by the agents employed in this system were described.

6.2 Challenges

In this thesis, the benefits and advantages of applying MAS technology in power system automation, particularly in protection, fault diagnosis and asset management have been described. It is also important to identify the key technical challenges that are yet to be overcome to allow the most effective implementation of MAS within the proposed power automation framework. These challenges are summarised in four areas, including agent platforms, learning, mobility and security, which are introduced as follows.

Platforms

Currently, a number of agent platforms exist, such as JADE mentioned in Chapter 1, providing an environment for agent execution and message transfer. However, one platform can not compatible with the others. It is necessary to develop the software agents that can interact with each other, irrespective of the platform they run on. Therefore, a flexible, extensible, robustness and open architecture for agent execution is required as a long-term objective for the development of agent-based power system automation.

Learning

More and more machine learning has been explored as a vital component to address challenges in MAS. Many researchers focused on making software agents to learn to cooperate with each other and human beings to achieve global objectives. Particularly, in power system automation, multi-agent learning faces significant challenges in understanding how agents can learn and adapt in the presence of other agents that are simultaneously learning and adapting [117]. For example, in the proposed agent-based transformer fault diagnosis system, a key challenge lies in the methodology, with which the rule base agent can train itself using the historical fault detection records to generate the more reliable and accurate rules automatically. In addition, other fields, such as Bayesian, game-theoretic, decision-theoretic, and evolutionary learning, can be also extended to more challenging multi-agent scenarios.

Mobility

Mobile agents that are able to move completely from a computer to another computer have been concerned by many researchers in recent years. As suggested in [47], the use of mobile agents may reduce the time consumption of agent communications in a high latency and low bandwidth network for remote control. In this case, another challenge in this framework is whether the proposed agent-based relaying schemes can utilise the mobile agents to improve the coordinations of the electric power equipments in the power networks. This is because power protection requires that the protection devices must operate rapidly if a fault occurs.

Security

Due to the peer-to-peer nature of a multi-agent system, security is a key concern that if an agent is seamlessly merged into a multi-agent system, the level of trust between agents and the security of the message transfer must be determined, because communication between two agents is open to attack, such as sender spoofing and message modification. In the proposed agent-based substation asset management system, this issue becomes more important, because power system operation may be incorrect and unsafe, if a vicious message is received and the agents send a wrong command to the combined power devices.

6.3 Suggestions for Future Work

Due to the broad scope of the research described in this thesis, insufficient time has been available to investigate all of the possibilities of a multi-agent framework for power system automation. This section addresses several related points that deserve further investigation. In some cases, the method by which these improvements might be achieved has been considered, but has not been implemented or tested in practice.

Embedded agents for real-time control and monitoring

The embedded agent should be investigated in future for the development of real-time control and monitoring in power systems. Real time considerations are an important factor in the design of a power system automation system. Therefore, the architecture should be enhanced to take account of these considerations, and an implementation including real-time behaviour should be created and evaluated. Furthermore, the embedded agents, or known as micro agents, could be developed that the software agents should be integrated with a micro chip, such as ARM, to communicate with other agents executed on PCs supporting remote access for real-time control and monitoring.

Ontology-driven agent reasoning

Ontology provides semantic annotations which are based on logic. The agents described in this thesis can be relatively limited. In fact, according to the FIPA agent specification, the agent behaviours are implemented in various standard types, *e.g.* simple behaviour, cyclic behaviour and parallel behaviour, *etc.* However, there should be another scope within the framework, especially the ontology inference and reasoning facility, to enable agents under this framework to be able to perform a wider variety of tasks in more "intelligent" way. Therefore, Ontology-driven agent reasoning should be proposed as a future work for the development of agents employed in this framework.

Bibliography

[1] D. J. Dolezilek. Power System Automation. Technical report, Schweitzer Engineering Laboratories, Inc., Pullman, WA, USA, 2000.

[2] J. L. Blackburn and T. J. Domin. *Protective Relaying: Principles and Applications*. CRC Press, New York, USA, 3rd edition, 2007.

[3] M. Öhrström. *Fast Fault Detection for Power Distribution Systems*. PhD thesis, Royal Institute of Technology (KTH) Stockholm, 2003.

[4] M. Geidl. Protection of power systems with distributed generation: State of the art. Technical report, Power Systems Laboratory, Swiss Federal Institute of Technology (ETH) Zurich, July 2005.

[5] *Network Protection and Automation Guide*. ALSTOM, 2002. ISBN 2951858906.

[6] V. Lohmann. New strategies for substation control, protection and access to information. In *Proceedings of the sixteenth international conference and exhibition on electricity distribution (CIRED 2001)*, volume 3, pages 211–215. IEE Publishing, 2001.

[7] National Asset Management Steering Group Australia. *International infrastructure management manual*. Association of Local Government Engineering New Zealand, Thames, New Zealand, 2002.

[8] R. Merritt. Asset management keeps plants running smarter. *Control (Chicago, III)*, 3(13):6 pages, 2000.

[9] L. Bertling, R. Allan, and R. Eriksson. A reliability-centered asset maintenance method for assessing the impact of maintenance in power distribution systems. *IEEE Transactions on Power Systems*, 1(20):75–82, 2002.

[10] J. Crisp and D. Birtwhistle. System dynamics modelling: application to electricity transmission network asset management. *Australian Journal of Electrical and Electronics Engineering*, 2(3):263–271, 2005.

[11] C. Hewitt. Viewing control structures as patterns of passing messages. *Artificial Intelligence*, 8(3):323–364, 1977.

[12] H. S. Nwana. Software agents: An overview. *Knowledge Engineering Review*, 11(3):1–40, 1996.

[13] L. Gasser, C. Braganza, and N. Herman. MACE: A flexible testbed for distributed AI research. In M. Huhns, editor, *Research Notes in Artificial Intelligence*, chapter 5, pages 119–152. London: Pitman, 1987.

[14] J. Doran, H. Carvajal, Y. Choo, and Y. Li. The MICS multi-agent testbed: Developments and experiments. In S. Deen, editor, *Cooperating Knowledge based Systems*, pages 240–251. Heidelberg: Springer-Verlag, 1991.

[15] K. S. Decker. TÆMS: A framework for environment centered analysis & design of coordination mechanisms. In *Foundations of Distributed Artificial Intelligence*, chapter 16. 1996.

[16] N. Carver and V. Lesser. The DRESUN testbed for research in FA/C distribution situation assessment: Extensions to the model of external evidence. In *Proceedings of the 1st International Conference on Multi-Agent Systems (ICMAS-95)*, pages 33–40, San Francisco, USA, 1991.

[17] N. R. Jennings, J. M. Corera, L. Laresgoiti, E. Mamdani, F. Perriollat, P. Skarek, and L. Varga. Using ARCHON to develop real-world DAI

applications for electricity transportation and particle accelerator control. *IEEE Expert, Special Issue on Real World Applications of DAI*, 1995.

[18] M. Wooldridge and N. R. Jennings. Intelligent agents: Theory and practice. *The Knowledge Engineering Review*, 10(2):115–152, 1995.

[19] M. Wooldridge and N. R. Jennings. *Intelligent Agents.* Heidelberg: Springer Verlag, 1995.

[20] S. Russell and P. Norvig. *Artificial Intelligence: A Modern Approach.* Prentice-Hall, 1995.

[21] N. R. Jennings. An agent-based approach for building complex software systems. *Communications of the ACM*, 44(4):35–41, 2001.

[22] J. Ferber. *Multi-Agent Systems : An Introduction to Distributed Artificial Intelligence.* ISBN 0201360489. Addison-Wesley, Harlow, England, 1999.

[23] R. A. Brooks. A robust layered control system for a mobile robot. *IEEE Journal of Robotics and Automation*, 2(1):14–23, 1986.

[24] R. Brooks. Intelligence without reason. In *Proceedings of the Twelfth International Joint Conference on Artificial Intelligence (IJCAI-91)*, Sydney, Australia, 1991.

[25] R. Brooks. Intelligence without representation. *Artificial Intelligence,*, 47, 1991.

[26] S. Rosenschein and L. Kaelbling. The synthesis of digital machines with provable epistemic properties. In J. Y. Halpern, editor, *Proceedings of the 1986 Conference on Theoretical Aspects of Reasoning About Knowledge*, pages 83–98. Morgan Kaufmann, 1986.

[27] M. J. Wooldridge and N. R. Jennings. Agent theories, architectures, and languages: A survey. *Lecture Notes in Artificial Intelligence*, 890:1–39, 1995.

[28] A. S. Rao and M. P. Georgeff. BDI agents: From theory to practice. In *Proceedings of the International Conference on Multi-Agent Systems (ICMAS)*, pages 312–319, San Francisco, USA, 1995.

[29] F. F. Ingrand, M. P. Georgeff, and A. S.: Rao. An architecture for real-time reasoning and system control. *IEEE Expert*, 7(6):34–44, 1992.

[30] M. P. Singh, A. S. Rao, and M. P. Georgeff. Formal methods in DAI: Logic-based representation and reasoning. In G. Weiss, editor, *Multiagent Systems: A Modern Approach to Distributed Artificial Intelligence*, pages 331–376. The MIT Press, Cambridge, Massachusetts, 1999.

[31] L. P. Kaelbling, M. L. Littman, and A. W. Moore. Reinforcement learning: A survey. *Journal of Artificial Intelligence Research*, 4:237–285, 1996.

[32] J. R. Koza, Bennett III, Andre F. H., and M. A. Keane. Genetic programming: Biologically inspired computation that creatively solves non-trivial problems. In L. Landweber, E. Winfree, R. Lipton, and S. Freeland, editors, *Proceedings of Discrete Mathematics and Theoretical Computer Science (DIMACS) Workshop on Evolution as Computation*, pages 15–44, 1999.

[33] S. Muggleton and L. Raedt. Inductive logic programming: Theory and methods. *Journal of Logic Programming*, 19/20:629–679, 1994.

[34] C. Olivia, C. F. Chang, C. F. Enguix, and A. K. Ghose. Case-based BDI agents: an effective approach for intelligent search on the world wide web. *AAAI Spring Symposium on Intelligent Agents*, 1999.

[35] P. Maes. Agents that reduce work and information overload. In J. M. Bradshaw, editor, *Software Agents*, chapter 8. MIT Press, Cambridge, Massachusetts, 1997.

[36] J. A. Boyan and M. L. Littman. Packet routing in dynamically changing networks: A reinforcement learning approach. In J. D. Cowan,

G. Tesauro, and J. Alspector, editors, *Advances in Neural Information Processing Systems*, pages 671–678. MIT Press, Cambridge, Massachusetts, 1993.

[37] P. Stone and M. Veloso. Multiagent systems: A survey from a machine learning perspective. *Autonomous Robotics*, 8(3), 2000.

[38] I. A. Ferguson. Touringmachines: Autonomous agents with attitudes. *IEEE Computer*, 25(5):51–55, 1992.

[39] K. Fischer, J. P. Muller, and M. Pischel. A pragmatic BDI architecture. In *Proceedings of ATAL 95, number LNAI 1037 in Lecture Notes in Artificial Intelligence*, pages 203–218. Springer Verlag, 1992.

[40] Foundation for Intelligent Physical Agents. fipa homepage. Available at `http://www.fipa.org`.

[41] Foundation for Intelligent Physical Agents (2000) FIPA agent standard specification. Available at `http://www.fipa.org/repository/standardspecs.html`.

[42] E. M. Davidson, S. D. J. McArthur, J. R.McDonald, T. Cumming, and I. Watt. Applying multi-agent system technology in practice: Automated management and analysis of SCADA and digital fault recorder data. *IEEE Transactions on Power Systems*, 21(2):559–567, May 2006.

[43] T. Nagata and H. Sasaki. A multi-agent approach to power system restoration. *IEEE Transactions on Power Systems*, 17(2):457–462, May 2007.

[44] S. E. Widergren, J. M. Roop, R. T. Guttromson, and Z. Huang. Simulating the dynamic coupling of market and physical system operations. In *IEEE Power Eng. Soc. General Meeting*, pages 748–753, June 2004.

[45] D. Koesrindartoto, S. Junjie, and L. Tesfatsion. An agent-based computational laboratory for testing the economic reliability of wholesale power market designs. In *IEEE Power Eng. Soc. General Meeting*, June 2005.

[46] A. L. Dimeas and N. D. Hatziargyriou. Operation of a multi-agent system for microgrid control. *IEEE Transactions on Power Systems*, 20(3):1447–1455, 2005.

[47] D. P. Buse, P. Sun, Q. H. Wu, and J. Fitch. Agent-based substation automation. *IEEE Power Energy Magazine*, 1(2):50–55, Mar./Apr. 2003.

[48] S. D. J. McArthur, E. M. Davidson, V. M. Catterson, A. L. Dimeas, N. D. Hatziargyriou, F. Ponci, and T. Funabashi. Multi-Agent Systems for Power Engineering Applications – Part I: Concepts, Approaches, and Technical Challenges. *IEEE Transactions on Power Systems*, 22(4):1743–1752, 2007.

[49] Foundation for Intelligent Physical Agents (FIPA), Available at http://www.fipa.org/. 2000.

[50] A. Korbik, S. D. J. McArthur, G. W. Ault, G. M. Burt, and J. R. McDonald. Enabling active distribution networks through decentralized autonomous network management. In *Proc. 18th Int. Conf. Electricity Distribution (CIRED)*. Turin, 2005.

[51] J. M. Corera, I. Laresgoiti, and N. R. Jennings. Using ARCHON. part 2: Electricity transportation management. *IEEE Expert*, 11(6):71–79, 1996.

[52] T. Nagata, H. Nakayama, M. Utatani, and H. Sasaki. A multi-agent approach to power system normal state operations. In *IEEE Power Engineering Society Summer Meeting*, volume 3, pages 1582–1586, 2002.

[53] J. M. Solanki, S. Khushalani, and N. N. Schulz. A multi-agent solution to distribution systems restoration. *IEEE Transactions on Power Delivery*, 22(3):1026–1034, Aug. 2007.

[54] S.-J. Park and J.-T. Lim. Modelling and control of agent-based power protection system using supervisors. In *Proceedings of Inst. Elect. Eng., Control Theory Appl.*, volume 153 of *1*, pages 92–98, 2006.

[55] R. Giovanini, K. Hopkinson, D. V. Coury, and J. S. Thorp. A primary and backup cooperative protection system based on wide area agents. *IEEE Transactions on Power Del.*, 21(3):1222–1230, 2006.

[56] I. S. Baxevanos and D. P. Labridis. Software agents situated in primary distribution networks: a cooperative system for fault and power restoration management. *IEEE Transactions on Power Delivery*, 22(4):2378–2385, Oct. 2007.

[57] E. E. Mangina, S. D. J. Mcarthur, and J. R. Mcdonald. Commas condition monitoring multi-agent system. *Autonomous Agents and Multi-agent Systems*, 4(3):279–282, 2001.

[58] E. E. Mangina, S. D. J. Mcarthur, and J. R. Mcdonald. Reasoning with modal logic for power plant condition monitoring. *IEEE Power Engineering Review*, 21(7):58–59, 2001.

[59] C. Lucas, M. A. Zia, M. R. A. Shirazi, and A. Alishahi. Development of a multi-agent information management system for iran power industry: A case study. In *Proceedings of 2001 IEEE Porto Power Tech Conference*, Porto, Portugal, 2001.

[60] V. Vishwanathan, J. Mccalley, and V. Honavar. A multiagent system infrastructure and negotiation framework for electric power systems. In *Proceedings of 2001 IEEE Porto Power Tech Conference*, volume 1, Porto, Portugal, 2001.

[61] V. Vishwanathan, V. Gangula, J. Mccalley, and V. Honavar. A multiagent systems approach for managing dynamic information and decisions in competitive electric power systems. In *NAPS*, Waterloo, Canada, 2000.

[62] S. E. Widergren, J. M. Roop, R. T. Guttromson, and Z. Huang. Simulating the dynamic coupling of market and physical system operations. In *IEEE Power Eng. Soc. General Meeting*, pages 748–753, 2004.

[63] K. Hopkinson, X. Wang, R. Giovanini, J. Thorp, K. Birman, and D. Coury. EPOCHS: A platform for agent-based electric power and communication simulation built from commercial off-the-shelf components. *IEEE Transactions on Power Systems*, 21(2):548–558, 2006.

[64] K. Caird. Integrating substation automation. *IEEE Spectrum*, 34(8):64–69, 1997.

[65] Z. Yang, C. Ma, J. Q. Feng, and Q. H. Wu. A multi-agent framework for power system automation. *International Journal of Innovations in Energy Systems and Power*, 1(1):39–45, 2006.

[66] Z. Yang, C. Ma, J. Q. Feng, and Q. H. Wu. A multi-agent framework for power system automation. In *Proc. of Artificial Intelligence in Energy Systems and Power (AIESP)*, Island of Madeira, Portugal, 2006.

[67] C. Ma, J. Q. Feng, Z. Yang, and Q. H. Wu. A broker agent for remote control of distributed power systems. In *Proc. of International Conference on Power System Technology*, Chongqing, China, 2006.

[68] Z. Yang, Z. Lu, C. Ma, Q. H. Wu, and J. Fitch. Improving control ability of relay protection system with intelligent agents. In *Proc. of International Conference on Power System Technology*, Chongqing, China, 2006.

[69] C. Ma, Z. Lu, W. H. Tang, and Q. H. Wu. An agent brokering-based scheme for anti-islanding protection of distribution networks. *Accepted by IEEE PES General Meeting*, 2006.

[70] C. Ma, W. H. Tang, Q. H. Wu, and J. Fitch. Asset managing the power dilemma. *IET Automation and Control Magazine*, pages 40–45, 2007.

[71] C. Ma, J. Q. Feng, Z. Yang, and Q. H. Wu. A transformer predictive maintenance system based on agent-oriented programming. In *Proc. of 2005 IEEE/PES Transmission and Distribution Conference and Exhibition: Asia and Pacific*, pages 1–6, Dalian, China, 2005.

[72] C. Ma, J. Q. Feng, Z. Yang, and Q. H. Wu. An agent-based personal article citation assistant. In *Proc. of IEEE International Joint Conference on Web Intelligence and Intelligent Agent Technology*, pages 720–705, Compiegne University of Technology, France, 2005.

[73] Q. H. Wu, D. P. Buse, P. Sun, and J. Fitch. An Architecture for e-Automation. *IEE Computing and Control Engineering*, 14(2):38–43, Apirl/May 2003.

[74] D. P. Buse and Q. H. Wu. *IP Network-based Multi-agent Systems for Industrial Automation.* ISBN: 9781846286469. Springer London, 2007.

[75] O. de Kretser, A. Moffat, T. Shimmin, and J. Zobel. Methodologies for distributed information retrieval. In *Proceedings of the Eighteenth International Conference on Distributed Computing Systems*, pages 66–73, Amsterdam, The Netherlands, 1998.

[76] F. Bellifemine, G. Caire, and D. Greenwood. *Developing Multi-Agent Systems with JADE.* ISBN: 9780470057476. John Wiley, U.K., 2007.

[77] F. Bellifemine, A. Poggi, and G. Rimassa. JADE - a FIPA-compliant agent framework. In *Proceedings of the 4th International Conference and Exhibition on The Practical Application of Intelligent Agents and Multi-Agents*, London, 1999.

[78] S. Poslad, P. Buckle, and R. Hadingham. The FIPA-OS agent platform: Open source for open standards. In *Proceedings of the 5th International Conference and Exhibition on the Practical Application of Intelligent Agents and Multi-Agents*, pages 355–368, Manchester, U.K., 2000.

[79] Y. Labrou, T. Finin, and Y. Peng. Agent communication languages: The current landscape. *IEEE Intelligent Systems*, 14(2):45–52, 1999.

[80] FIPA SL content language specification. Available at `http://www.fipa.org/specs/fipa00008/`.

[81] F. Bergenti and A. Poggi. Leap: A fipa platform for handheld and mobile devices. In *Intelligent agents*, volume VIII, pages 436–445. Springer-Verlag, Berlin Heidelberg, 2002.

[82] R. G. Smith and R. Davis. Frameworks for cooperation in distributed problem solving. *IEEE Transactions on Systems, Man and Cybernetics*, 11(1):61–69, 1981.

[83] M. Nodine, A. Ngu, A. Cassandra, and W. G. Bohrer. Scalable semantic brokering over dynamic heterogeneous data sources in infosleuth. *IEEE Transactions on Knowledge and Data Engineering*, 15(5):1082–1098, September/October 2003.

[84] Interactive Maryland Platform for Agents Collaborating Together. Available at http ://www.cs.umd.edu/projects/impact.

[85] K. Sycara and D. Zeng. Coordination of multiple intelligent software agents. *Cooperative Information Systems*, 5(2/3), 1996.

[86] Available at `http://www.cs.cmu.edu/softagents/retsina.html`.

[87] FIPA Brokering Interaction Protocol Specification. Available at `http://www.fipa.org/specs/fipa00033`.

[88] D. Birla, R. P. Maheshwari, and H. O. Gupta. An approach to tackle the threat of sympathy trips in directional overcurrent relay coordination. *IEEE Transactions on Power Delivery*, 22(2):851–858, 2007.

[89] T. E. McDermott and R. C. Dugan. Distributed generation impact on reliability and power qualityindices. In *Proceedings of Rural Electric Power Conference*, pages D3–D3–7, 2002.

[90] M. T. Doyle. Reviewing the impacts of distributed generation on distribution system protection. In *Proceedings of IEEE Power Engineering Society Summer Meeting Conference*, volume 1, pages 103–105, Jul. 2002.

[91] Distribution power generator islanding guidelines. Technical report, British Columbia hydro, 2006.

[92] N. Jenkins and et al. *Embedded generation.* ISBN 0-85296-774-8. IEE, 2000.

[93] The potential benefits of distributed generation and rate-related issues that may impede its expansion. Technical report, U.S. Department of Energy, Jun. 2007.

[94] P. P. Barker. Determining the impact of distributed generation on power systems: Part 1-radial distribution systems. In *Proceedings of IEEE Power Eng. Soc. Summer Meeting*, pages 1645–1656, 2000.

[95] L. A. Kojovic and L. D. Willoughby. Integration of distributed generation in a typical usa distribution system. In *Proceedings of the 16th Int. Conf. Exhib. Electricity Distribution (CIRED)*, volume 4, 2001.

[96] IEEE Extremely Inverse TOC Curves. Available at http://www.geindustrial.com/, 2008.

[97] M. Öhrström. *Fast fault detection for power distribution systems.* PhD thesis, Royal Institute of Technology (KTH) Stockholm, 2003.

[98] G. Hodgkingson. System implications of embedded generation and its protection and control. Technical report, IEE, 1998.

[99] I. S. Baxevanos and D. P. Labridis. Implementing multiagent systems technology for power distribution network control and protection management. *Implementing Multiagent Systems Technology*, 22(1):433–443, 2007.

[100] H. Wan, K. K. Li, and K. P. Wong. Multi-agent application of substation protection coordination with distributed generators. *European Transactions on Electrical Power*, 16:495–506, 2006.

[101] IEEE Guide for Interfacing Dispersed Storage and Generation Facitlities With Electric Utility Systems. *IEEE/ANSI Std. 1001-1988.*

[102] S. I. Jang and K. H. Kim. An islanding detection method for distributed generations using voltage unbalance and total harmonic distortion of current. *IEEE Transactions on Power Delivery*, 19(2):745–752, 2004.

[103] J. Stevens, R. Bonn, J. Ginn, and S. Gonzalez. Development and testing of an approach to anti-islanding in utility-interconnected photovoltaic systems. Technical Report SAND 2000-1939, Photovoltaic System Applications Department Sandia National Laboratories, 2000.

[104] M. Guillot and et al. Protection of embedded generation connected to a distribution network and loss of mains detection. In *Proceedings of the 16th International Conference and Exhibition on Electricity Distribution*, volume 4, page 5, 2001.

[105] P. O'Kane and B. Fox. Loss of mains detection for embedded generation by system impedance monitoring. In *Proceedings of Sixth International Conference on Developments in Power System Protection*, pages 95–98, 1997.

[106] O. Usta, M. A. Redfern, and J. I. Barrett. Protection of dispersed storage and generation units against islanding. In *Proceedings of the 7th Mediterranean Electrotechnical Conference*, volume 3, pages 976–979, 1994.

[107] Z. Ye et al. Testing of GE universal interconnection device. Technical Report NREL/TP-560-34676, National Renewable Energy Laboratory, 2003.

[108] A. Dyśko, C. Booth, O. Anaya-Lara, and G. M. Burt. Reducing unnecessary disconnection of renewable generation from the power system. *IET Renew. Power Generation*, 1(1):41–48, 2007.

[109] Z. Ye. Study and development of anti-islanding control for synchronous machine-based distributed generators. Technical report, GE Global Research Center, Niskayuna, New York, 2001.

[110] G. A. Smith, P. A. Onions, and D.G. Infield. Predicting islanding operation of grid connected PV inverters. *IEE Proceedings on Electric Power Applications*, 147(6):1–6, 2000.

[111] O. Tsukamoto, T. Okayasu, and K. Yamagishi. Study on islanding of dispersed photovoltaic power systems connected to a utility power grid. *Solar Energy*, 70(6):505–511, 2001.

[112] W. Freitas, Z. Huang, and W. Xu. A Practical Method for Assessing the Effectiveness of Vector Surge Relays for Distributed Generation Applications. *IEEE Transactions on Power Delivery*, 20(1):57–63, 2005.

[113] G. H. González and R. Iravani. Current injection for active islanding detection of electronically-interfaced distributed resources. *IEEE Transactions on Power Delivery*, 21(3):1698–1705, 2006.

[114] W. Xu, G. Zhang, W. Wang, G. Wang, and J. Kliber. A Power Line Signaling Based Technique for Anti-Islanding Protection of Distributed GeneratorsPart I: Scheme and Analysis. *IEEE Transactions on Power Delivery*, 22(3):1758–1766, 2007.

[115] J. Schneider, A. J. Gaul, and C. Neumann. Asset management techniques. *Electrical Power and Energy Systems*, 28:643–654, 2006.

[116] A. A. Freitas. A survey of evolutionary algorithms for data mining and knowledge discovery. In *A. Ghosh and S. Tsutsui. (Eds.) Advances in Evolutionary Computation*, pages 819–845. Springer-Verlag, August 2002.

[117] Workshop on multi-agent learning: Theory and practice, 2000. Stanford University.